



SVEUČILIŠTE U MOSTARU
FAKULTET STROJARSTVA RAČUNARSTVA I ELEKTROTEHNIKE

Programiranje u Javi

*Projektni zadatak: **Airline Database System***

Antonio Pavković

Akadska godina 2020/2021.

PROJEKTNI ZADATAK:

Opis problema:

Detaljno opisati problem koji je potrebno riješiti izgradnjom programske podrške. Svaki student piše svoju temu minimalno 500 znakova.

Ovaj softver upravlja i distribuira važne zrakoplovne dokumente. Sustav se koristi za kontrolu distribucije dokumenata određenim skupinama ili pojedincima unutar zrakoplovne tvrtke.

Projekt pomaže u upravljanju svim vašim podacima kako bi se pravilno i učinkovito kontrolirali resursi.

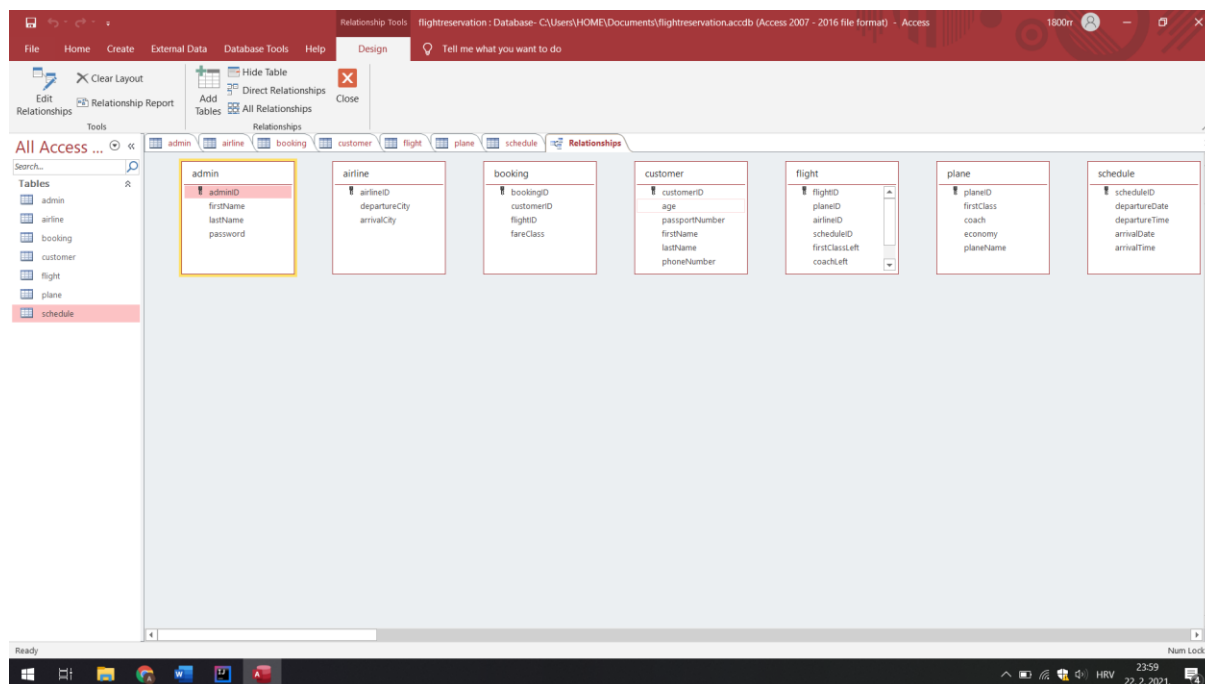
Cilj ovog projekta je olakšavanje rezervacije karte, dodavanja i uređivanja kupaca, letova/ruta, zrakoplova i rezervacija. Dizajnom baze podataka i implementiranim kodom osigurano je brzo i efikasno rezerviranje karte. Također podaci o kupcima su osigurani.

1. Modeliranje podataka

Prvi dio zadatka se radi u .

- Poglavlje 1.1. (Relacijski model podataka) - potrebno je staviti sliku relacijskog modela kreiranog u MS Accessu.
- Poglavlje 1.2. (Opis tablica u relacijskom modelu) - potrebno je navesti i opisati sve tablice (i atribute unutar tablica) iz dobivenog modela te na kraju priložiti sliku
- Poglavlje 1.3. (Aplikacija u Javi) - sadrži screenshotove kreiranih formi Javi.

1.1. Relacijski model podataka



1.2. Opis tablica u relacijskom modelu

Admin

Naziv atributa	Integritet (PK/FK, NULL / NOT NULL)	Kratki opis atributa
adminID	PK NOT NULL Auto Increment	ID admina
firstName	NULL	Ime
lastName	NULL	Prezime
Password	NULL	Lozinka

Airline

Naziv atributa	Integritet (PK/FK, NULL / NOT NULL)	Kratki opis atributa
----------------	--	----------------------

airlineID	PK NOT NULL Auto Increment	ID zrakoplovne kompanije
departureCity	NULL	Odlazni grad
arrivalCity	NULL	Dolazni grad

Booking

Naziv atributa	Integritet (PK/FK, NULL/NOT NULL)	Kratki opis atributa
bookingID	PK NOT NULL Auto Increment	ID rezervacije
customerID	NULL	ID kupca
flightID	NULL	ID leta
fareClass	NULL	Vrsta klase(first, coach, economy)

Customer

Naziv	Integritet (PK/FK, NULL/NOT NULL)	Kratki opis atributa
customerID	PK NOT NULL Auto Increment	ID kupca
age	NULL	Dob
passportNumber	NULL	Broj putovnice
firstName	NULL	Ime
lastName	NULL	Prezime
phoneNumber	NULL	Broj telefona/mobitela

Flight

Naziv	Integritet (PK/FK, NULL/NOT NULL)	Kratki opis atributa
flightID	PK NOT NULL Auto Increment	ID leta
planeID	NULL	ID aviona
airlineID	NULL	ID zrakoplovne kompanije
scheduleID	NULL	ID rasporeda
firstClassLeft	NULL	Ostatak sjedana na prvoj klasi
coachLeft	NULL	Ostatak sjedana na coach klasi
economyLeft	NULL	Ostatak sjedana na economy klasi
price	NULL	Cijena

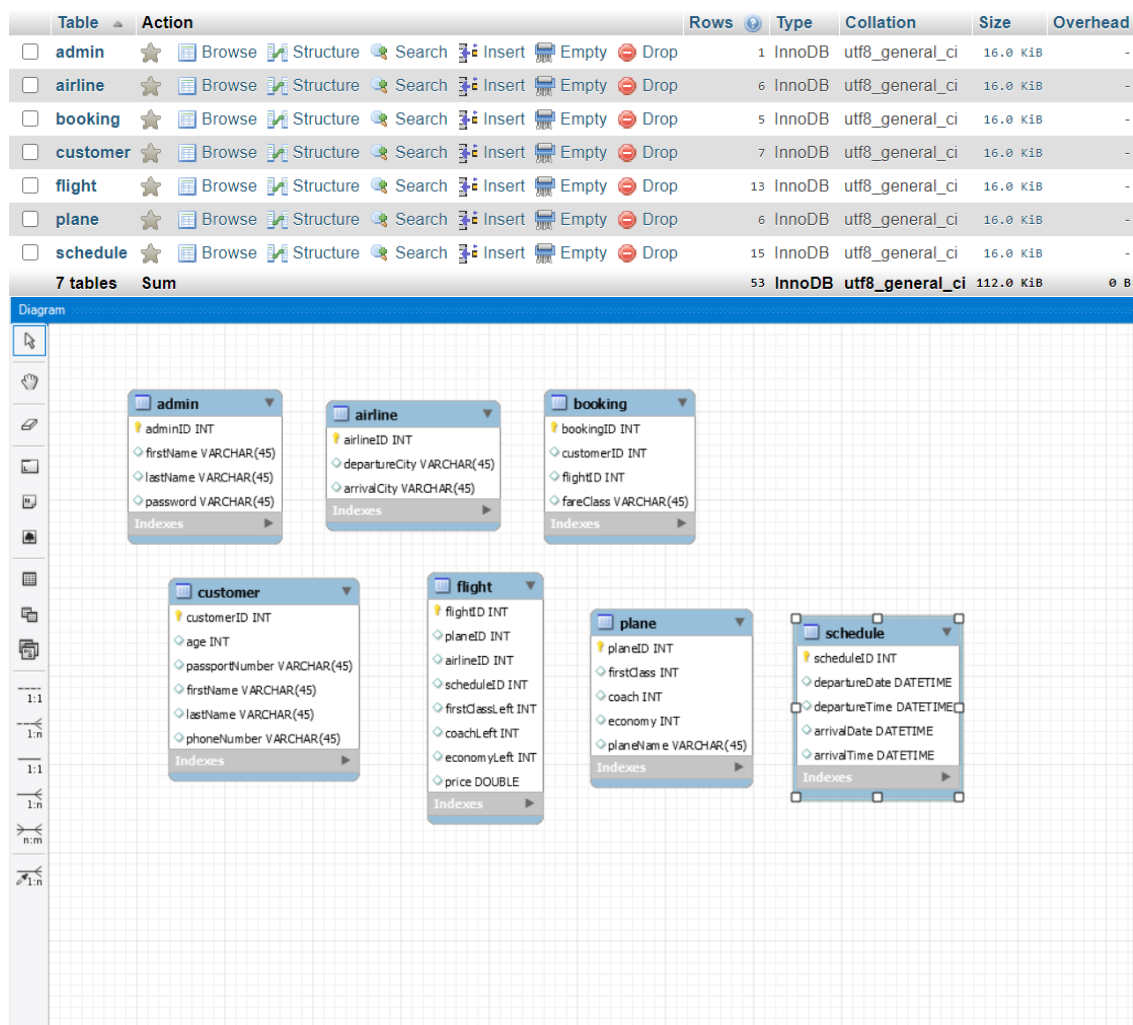
Plane

Naziv	Integritet (PK/FK, NULL/NOT NULL)	Kratki opis atributa
planeID	PK NOT NULL Auto Increment	ID aviona
firstClass	NULL	Broj sjedala na prvoj klasi
coach	NULL	Broj sjedala na coach klasi
economy	NULL	Broj sjedala na economy klasi
planeName	NULL	Ime aviona

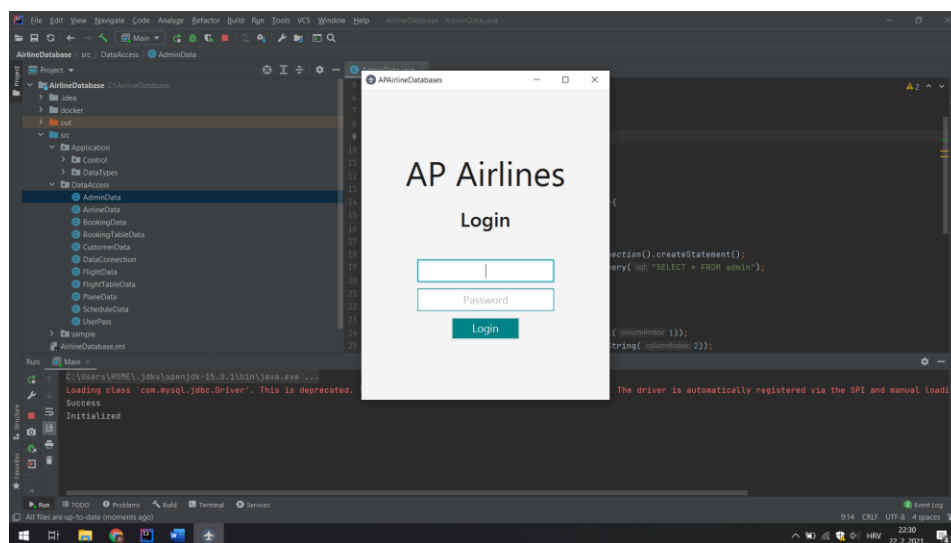
Schedule

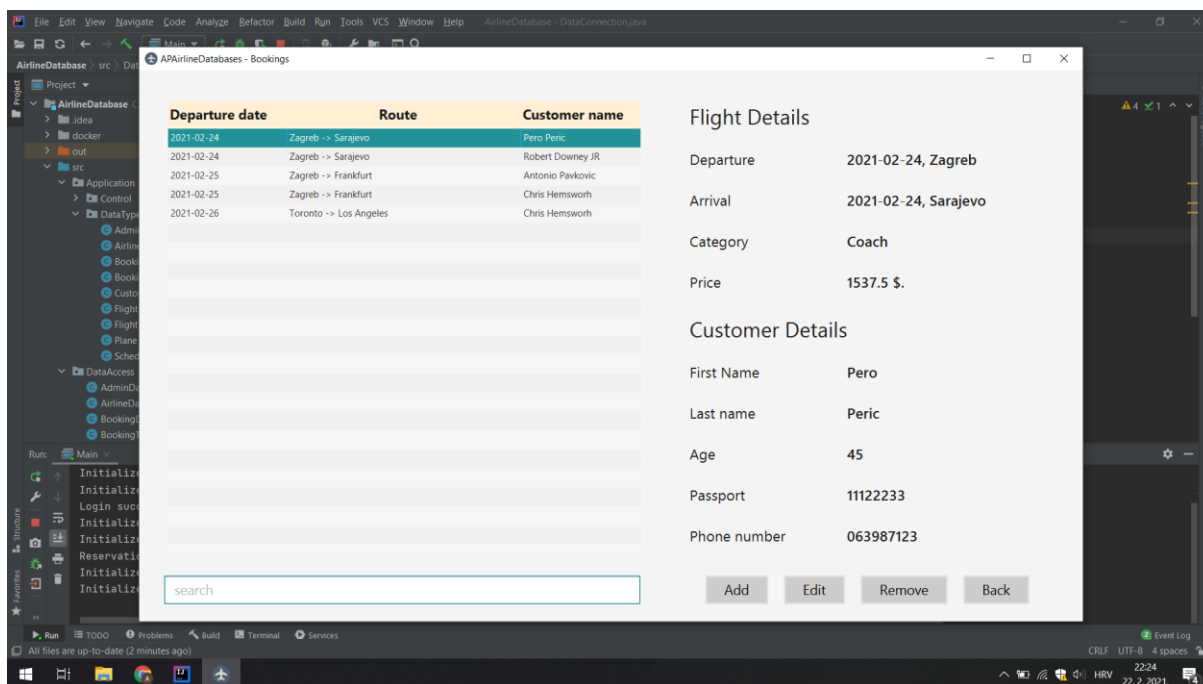
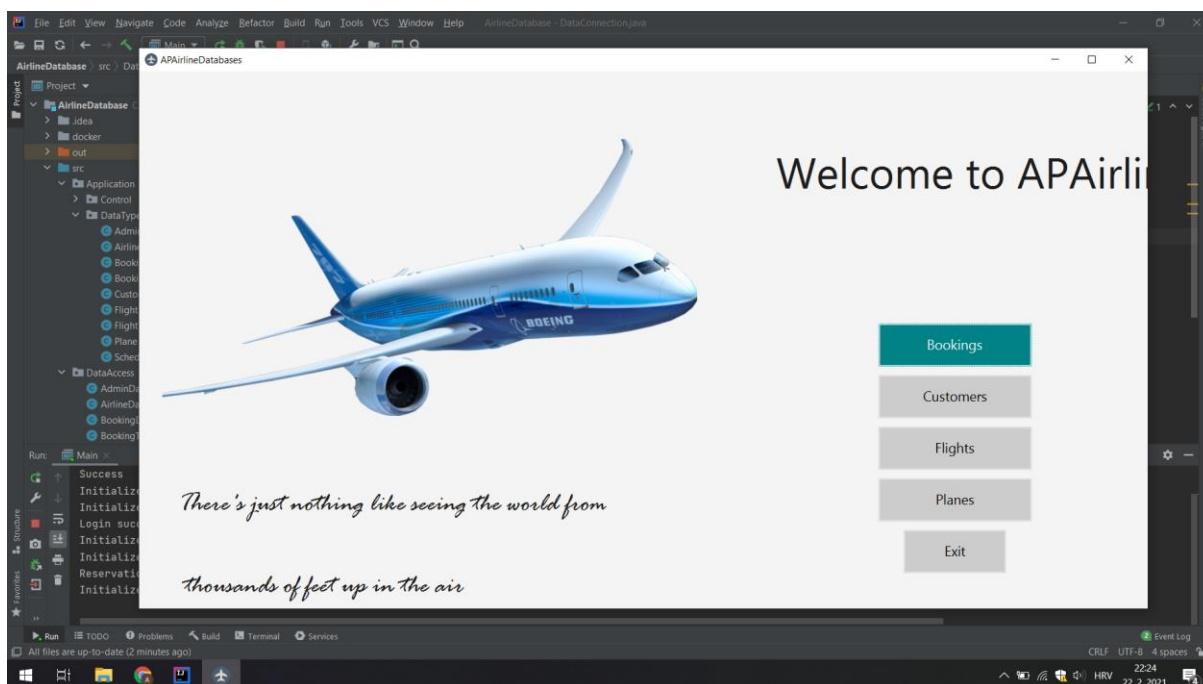
Naziv	Integritet (PK/FK, NULL/NOT NULL)	Kratki opis atributa
scheduleID	PK NOT NULL Auto Increment	ID rasporeda
departureDate	NULL	Datum polaska
departureTime	NULL	Vrijeme polaska
arrivalDate	NULL	Datum dolaska

arrivalTime	NULL	Vrijeme dolaska
-------------	------	-----------------

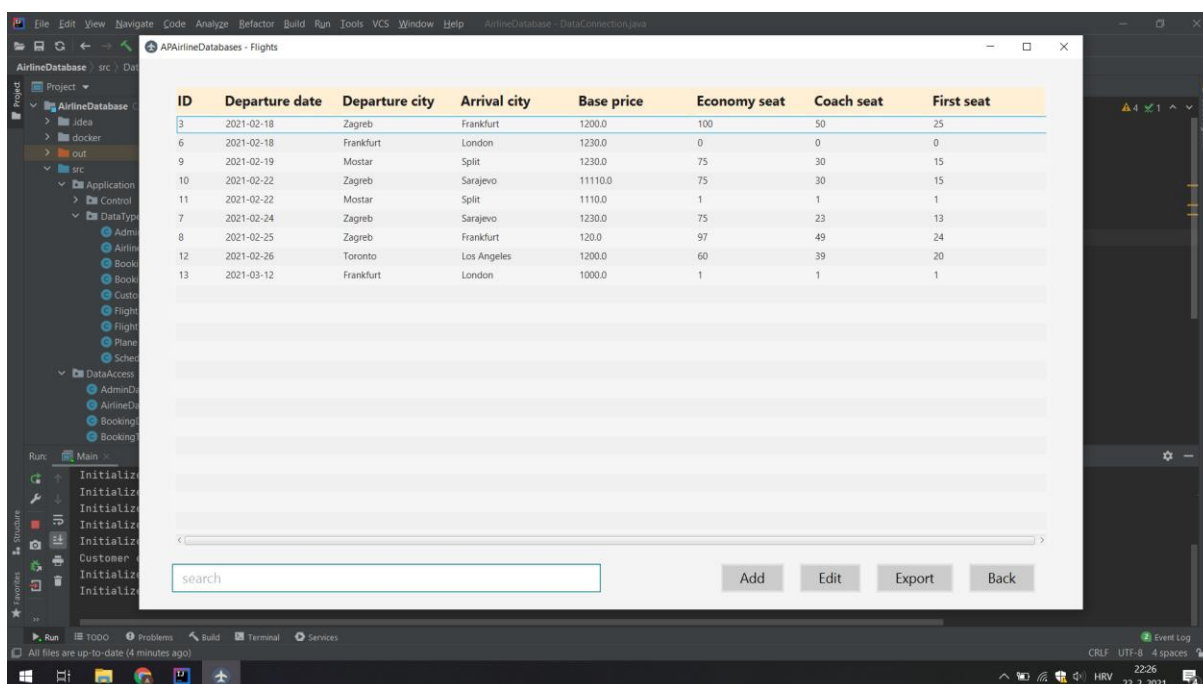
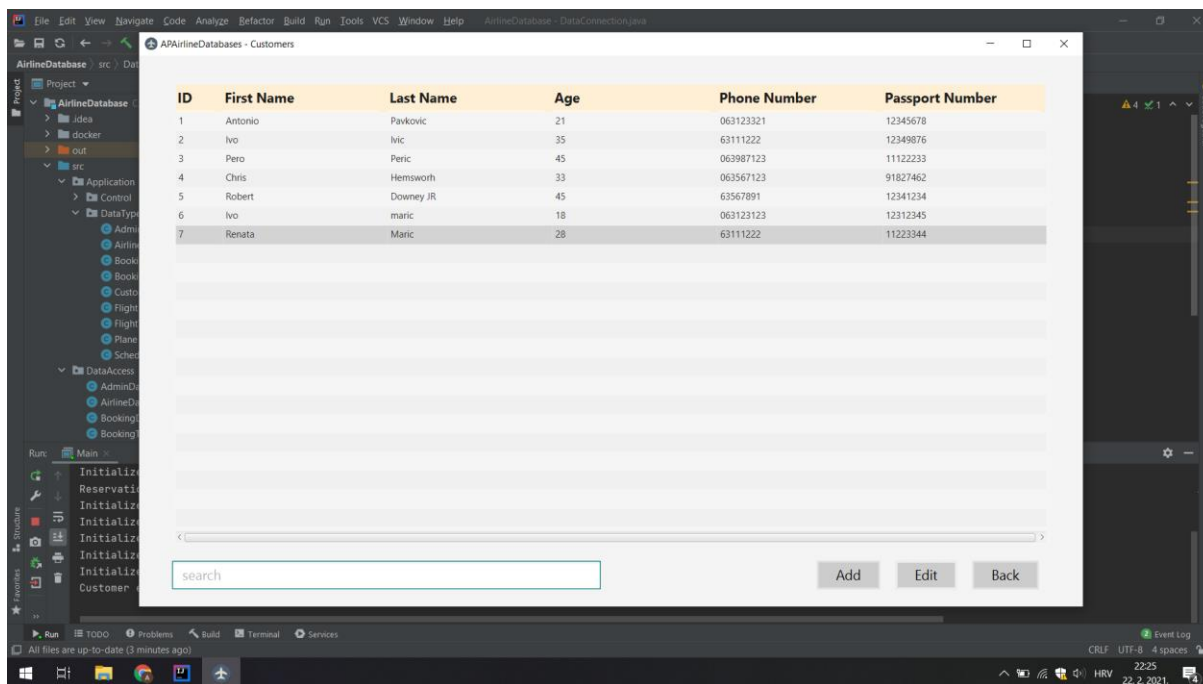


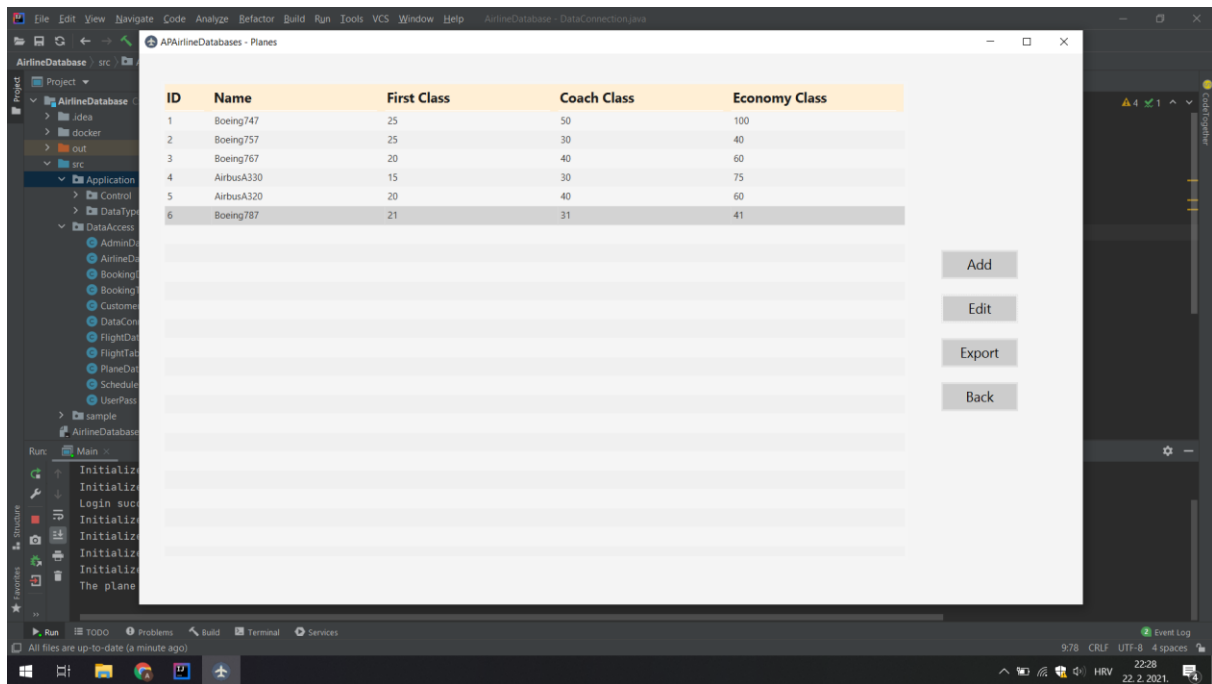
1.3. Aplikacija u Javi





Tehnička dokumentacija projekta





Tehnička dokumentacija projekta

APAirlineDatabases - Bookings

Departure date	Route
2021-02-24	Zagreb -> Sarajevo
2021-02-24	Zagreb -> Sarajevo
2021-02-25	Zagreb -> Frankfurt
2021-02-25	Zagreb -> Frankfurt
2021-02-26	Toronto -> Los Angeles
2021-02-26	Toronto -> Los Angeles
2021-02-26	Toronto -> Los Angeles
2021-03-14	Mostar -> Split

search

AddEditRemoveBack

Booking menu

Booking details

Select customer

OR

Route

Departure date

Category

Price

OKCancel

APAirlineDatabases - Customers

ID	First Name	Last Name	Passport Number
1	Antonio	Pavko	12345678
2	Ivo	Ivic	12349876
3	Pero	Peric	11122233
4	Chris	Hemsley	91827462
5	Robert	Downing	12341234
6	Ivo	maric	12312345
7	Renata	Maric	11223344
8	Marko	Markovic	12341234

search

AddEditBack

Customer menu

Customer Info

First name

Last name

Age

Phone number

Passport number

OkCancel

APAirlineDatabases - Flights

ID	Departure date	Departure
3	2021-02-18	Zagreb
6	2021-02-18	Frankfurt
9	2021-02-19	Mostar
10	2021-02-22	Zagreb
11	2021-02-22	Mostar
7	2021-02-24	Zagreb
15	2021-02-25	Zagreb
8	2021-02-25	Zagreb
12	2021-02-26	Toronto
14	2021-03-05	Toronto
13	2021-03-12	Frankfurt
16	2021-03-14	Mostar

search

AddEditExportBack

Flight menu

Flight details

Route

->

Date

27. 2. 2021.

Time

7:00

To

Plane ID

0

Base price

0.0

Ok

Cancel

Coach seat

First seat

50	25
0	0
30	15
30	15
1	1
23	13
40	20
49	24
38	19
30	25
1	1
39	20

APAirlineDatabases - Planes

ID	Name	First
1	Boeing747	25
2	Boeing757	25
3	Boeing767	20
4	AirbusA330	15
5	AirbusA320	20
6	Boeing787	21
7	AirbusA333	15

Plane menu

Plane details

Plane name

First class

0

Coach class

0

Economy class

0

Ok

Cancel

Add

Edit

Export

Back

2. Model programske podrške

Drugi dio zadatka se odnosi na modeliranje programske podrške, detaljno ćete raspisati sve klase i na kraju priložiti dijagram klase koji opisuje izgled programske podrške:

- Napraviti class dijagram
- Opisati metode koje se nalaze unutar svake klase

U navedenim klasama potrebno je opisati :

- 1) Paketi – kratko opisati paket vaše programske podrške
- 2) Klase – kratko opisati klase koje koristite (ukratko za što će se klasa koristiti
 - a. Atribute – kratko pisati atribute koje se nalaze unutar klase;
 - b. Metode – kratko opisati metode koje se koriste za što služe i opišite njihov povratni tip;

2.1. Paketi

Opis paketa

Projekt Airline Database System sastoji se od tri paketa: Application, DataAccess i sample.

Paket Application sastoji se od dva paketa: Control i DataTypes.

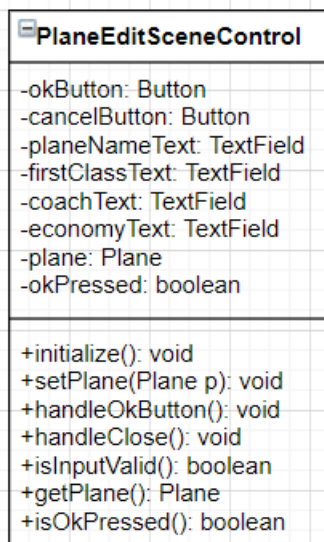
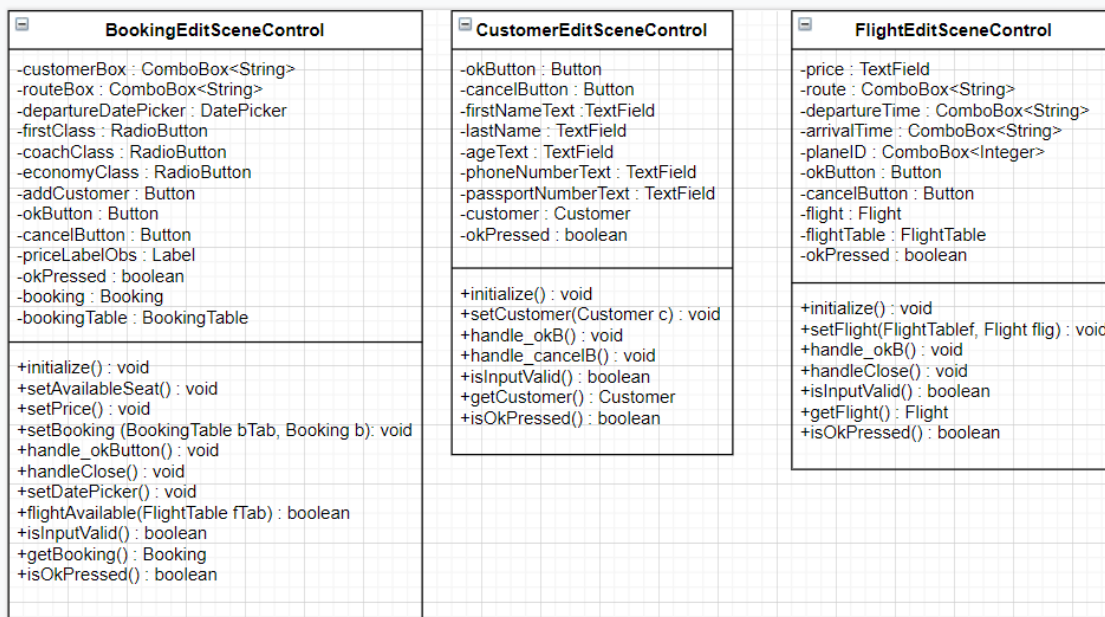
Application: Control se sastoji od 11 klasa u kojima se inicijaliziraju sporedni prozori/forme, također se dohvaćaju podaci iz baze, računa povrat novca prilikom otkazivanja rezervacije i ispis osnovnih podataka o letovima i kupcima

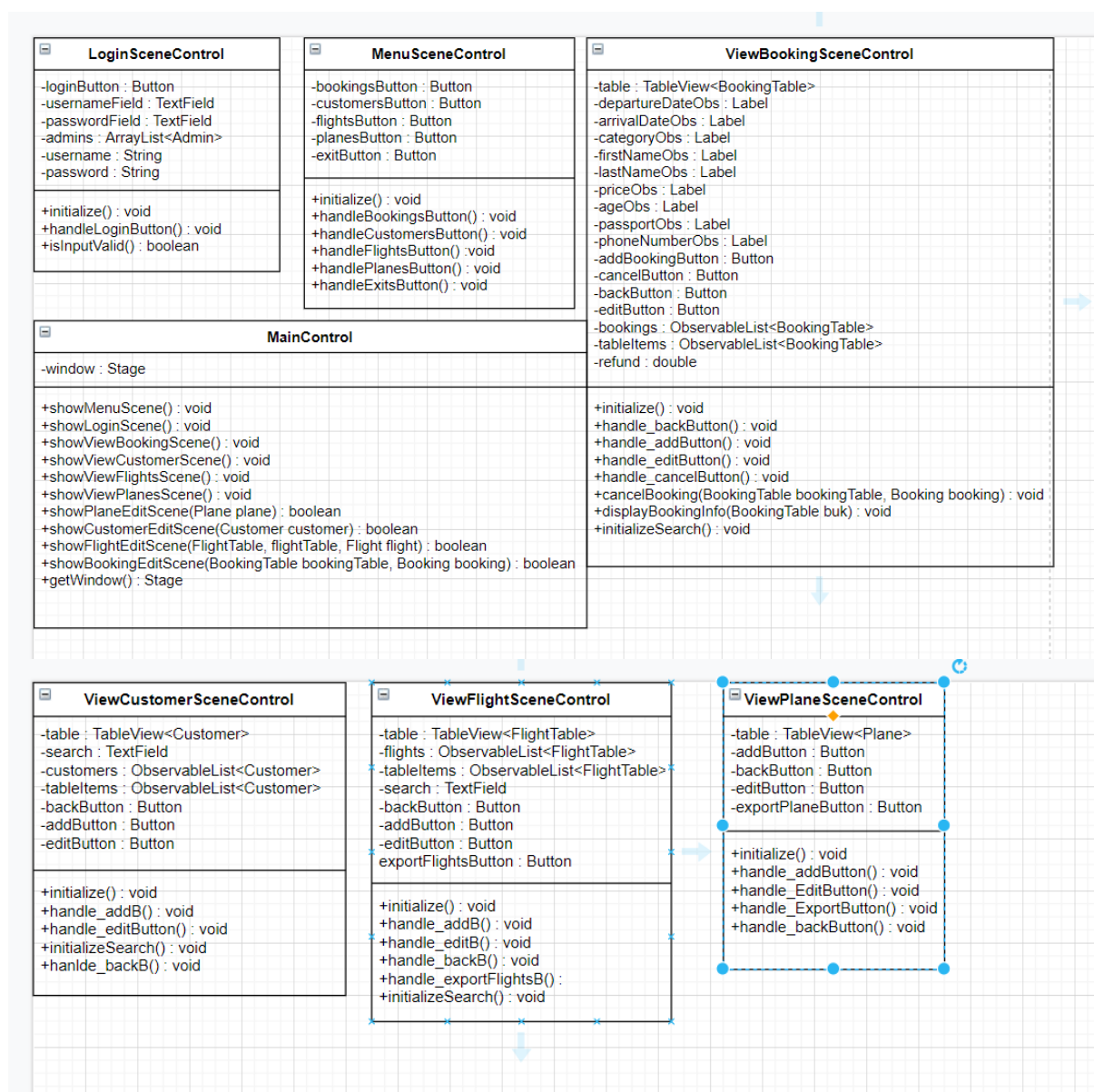
Application: DataTypes se sastoji od 9 klasa kojima se kreiraju objekti kojima se kreiraju tipovi podataka vezani za bazu

DataAccess: se sastoji od 11 klasa pomoću kojih se radi sa bazom podataka i spaja na bazu

sample: se sastoji od 11 klasa kojima se inicijaliziraju glavni „frontend“ dijelovi aplikacije. Također se u njoj nalazi Main klasa koja sadrži start metodu.

2.2. Klase - Application: Control





2.2.1. Klasa 1 - BookingEditSceneControl

Klasa se koristi za inicijalizaciju i funkcionalnost „pomoćnog prozora“ koji se pojavi prilikom rezervacije novoga leta.

Metoda dohvaća sve kupce i letove koje postoje u bazi, prati broj slobodnih mjesta i u slučaju da nema mjesta onemogućava gumb za dodavanje novih putnika. Također klasa definira početnu cijenu karte i računa cijenu na osnovu klase u kojoj kupac želi sjediti (first class, coach ili economy). Unutar metode su definirane funkcionalnosti za gumb „Ok“ i „Close“, te mogućnosti odabira datuma leta. BookingEditFlightScene također ima metodu za provjeru unosa podataka, tj. javlja grešku ako nismo odabrali putnika ili rutu.

2.2.1.1. Atributi

private static ComboBox<String> customerBox – atribut za dodavanje kupca
private static ComboBox<String> routeBox – atribut za dodavanje rute

private static DatePicker departureDatePicker – atribut za odabir datuma polaska
private static RadioButton firstClass – radio gumb za odabir „first class“
private static RadioButton coachClass – radio gumb za odabir „coach class“
private static RadioButton economyClass – radio gumb za odabir „economy class“
private static Button addCustomer – gumb za dodavanje kupca
private static Button okButton – gumb „Ok“
private static Button cancelButton – gumb „Cancel“
private static Label priceLabelObs – atribut pomocu kojeg se prikazuje cijena rezervirane leta
private static boolean okPressed = false – boolean atribut kojim pratimo ako je gumb „OK“ pritisnut odnosno ako nije
private static Booking booking - atribut za rad s bazom
private static BookingTable bookingTable - atribut za rad s bazom

2.2.1.2. Metode

public static void initialize() – inicijalizacija prozora, dohvaća kupce i rute i postavlja gumbove na odgovarajuća mjesta

public static void setAvailableSeat() – metoda za onemogućivanje gumba u slučaju da nema slobodnih mjesta

public static void setPrice() – metoda za postavljanje cijene leta u ovisnosti odabane klase (first, coach, economy)

public static void setBooking(BookingTable bTab, Booking b) – metoda za postavljanje rezervacije za uređivanje

public static void handle_okButton() – funkcionalnost gumba „Ok“ , ako je formular popunjen let se rezervira, inače javlja grešku

public static void handleClose() – metoda koja zatvara prozor za rezervaciju

public static void setDatePicker() – metoda koja postavlja kalendar i omogućuje odabir datuma za let

public static boolean flightAvailable(FlightTable fTab) – metoda koja provjerava ima li slobodnih mjesta na letu, vraća true ili false

public static boolean isValid() – metoda koja provjerava je li popunjen formular i/ili ima li grešaka prilikom popunjavanja formulara za rezervaciju. Vraća true ili false

public static Booking getBooking() – metoda(getter) koja vraća rezervaciju

public static boolean isOkPressed() – metoda koja vraća je li gumb „Ok“ pritisnut. Vraća true ili false

2.2.2 Klasa 2 – CustomerEditSceneControl

Klasa se koristi za inicijalizaciju i funkcionalnost „pomoćnog prozora“ koji se pojavi prilikom dodavanja novog kupca.

Ovo je jednostavni formular u kojem se jednostavno mogu dodati osnovni podaci o kupcu, tipa ime, prezime, dob, broj telefona/mobitela i putovnice.

Takodjer imaju metode za funkcionalnost gumbova „Ok“ i „Cancel“, te metoda za provjeru unosa u formular

2.2.2.1. Atributi klase 2

private static Button okButton, cancelButton – atributi za gumbove „Ok“ i „Cancel“
private static TextField firstNameText, lastNameText, ageText, phoneNumberText, passportNumberText – atributi-polja za unos teksta za ime, prezime, dob, broj telefona i putovnice
private static Customer customer = new customer() – atribut za inicijalizaciju novog objekta tipa Customer
private static boolean okPressed = false – boolean atribut kojim pratimo ako je gumb „OK“ pritisnut odnosno ako nije

2.2.2.2. Metode klase 2

public static void initialize() – inicijalizacija prozora, potavlja gumbove i polja za unos teksta na odgovarajuća mjesta

public static void setCustomer(Customer c) – setter, takodjer postavlja tekst unutar formulara

public static void handle_okB() - funkcionalnost gumba „Ok“ , ako je formular popunjen kupac se dodaje u bazu, inače javlja grešku

public static void handle_cancelB() - metoda koja zatvara prozor

public static void isValid() – metoda za provjeru korisničkog unosa. Vraća true ili false

public static void Customer getCustomer() -metoda(getter) koja vraća kupca

public static boolean isOkPressed() - metoda koja vraća je li gumb „Ok“ pritisnut. Vraća true ili false

2.2.3 Klasa 3 – FlightEditSceneControl

Klasa se koristi za inicijalizaciju, funkcionalnost, dodavanje i uređivanje letova koje korisnik pretražuje, te njihovih cijena i aviona koji se koriste za let. Također je moguće urediti i vrijeme i datum polaska i dolaska.

Također su definirane funkcionalnosti gumbova „Ok“ i „Cancel“, te metoda za provjeru korisnikova unosa.

2.2.3.1. Atributi klase 3

public static TextField price – polje za unos teksta/cijene
public static ComboBox<String> route, departureTime, arrivalTime – atributi za dodavanje/odabir rute, vremena polaska i dolaska

public static ComboBox<Integer> planeID – atribut za odabir identifikacijskog broja aviona
public static DatePicker departureDate – atribut za odabir datuma polaska
public static Button okButton, cancelButton – gumbovi „Ok“ i „Cancel“
public static Flight flight – atribut za dohvaćanje letova iz tablice/rad s bazom
public static FlightTable flightTable - atribut za dohvaćanje letova iz tablice/rad s bazom
public static boolean isOkPressed = false – boolean atribut kojim pratimo ako je gumb „OK“ pritisnut odnosno ako nije

2.2.3.2. Metode klase 3

public static void initialize() – inicijalizacija prozora, potavlja gumbove i polja za unos teksta na odgovarajuća mjesta i dohvaća iz baze sve rute za let, identifikacijske brojeve aviona i rad s datumom i vremenom polaska
public static void setFlight(FlightTable f, Flight flig) – metoda za postavljanje letova za uređivanje

public static void handle_okB() - funkcionalnost gumba „Ok“ , ako je formular popunjen kupac se dodaje u bazu, inače javlja grešku

public static void handle_cancelB() - metoda koja zatvara prozor

public static void isValid() – metoda za provjeru korisničkog unosa. Vraća true ili false

public static void Flight getFlight() -metoda(getter) koja vraća let

public static boolean isOkPressed() - metoda koja vraća je li gumb „Ok“ pritisnut. Vraća true ili false

2.2.4 Klasa 4 – LoginSceneControl

Klasa se koristi za inicijalizaciju i funkcionalnost login ekrana. Također iz baze se provjeravaju korisnički podaci za logiranje. Definirane su metode za login gumb za login i metoda za provjeru korisničkog unosa

2.2.4.1. Atributi klase 4

private static Button loginButton – atribut tipa Button za login gumb
private static TextField usernameField – atribut/polje za unos teksta za korisničko ime
private static TextField passwordField – atribut/polje za unos teksta za lozinku
private static ArrayList<Admin> admins atribut tipa ArrayList koja služi kao lista admina
private static String username – atribut za korisničko ime
private static String password -atribut za lozinku

2.2.4.2. Metode klase 4

public static void handleLoginButton() – metoda za funkcionalnost login gumba, provjera korisničkih podataka u bazi, ako su podaci točni funkcija poziva glavni prozor aplikacije, a ako su podaci netočni metoda izbacuje grešku.

public static void isValid() – metoda za provjeru korisničkog unosa. Vraća true ili false

2.2.5 Klasa 5 – MainControl

Klasa se koristi za inicijalizaciju i „povezivanje“ ostalih dijelova projekta

2.2.5.1. Atributi klase 5

public static Stage window = new Stage() – objekt tipa Stage, koristi se kao atribut na kojemu se realizira i konstruira projekt

2.2.5.2. Metode klase 5

public static void showMenuScene() – inicijalizacija glavnog prozora

public static void showLoginScene() – inicijalizacija login prozora i desktop ikone, te povezivanje s bazom

public static void showBookingScene() -inicijalizacija prozora za rezervacije letova

public static void showViewCustomerScene() – inicijalizacija prozora za pregled kupaca

public static void showViewFlightScene() – inicijalizacija prozora za pregled letova

public static void showViewPlaneScene() – inicijalizacija prozora za pregled aviona

public static boolean showViewPlaneEditScene() – inicijalizacija prozora za uređivanje letova, metoda vraća uređen let ili grešku

public static boolean showCustomerEditScene(Customer) – inicijalizacija prozora za uređivanje kupaca, metoda vraća uređenog kupca ili grešku

public static boolean showFlightEditScene(FlightTable flightTable, Flight flight) – inicijalizacija metode za uređivanje letova

public static boolean showBookingEditScene(BookingTable bookingTable, Booking booking) - inicijalizacija metode za uređivanje rezervacije

public static Stage getWindow() – getter metoda za window

2.2.6 Klasa 6 – MenuSceneControl

Klasa za inicijalizaciju glavnog izbornika. Klasa postavlja gumbove, sliku pozadine, naslov i i izreku na odgovarajuća mjesta. Također gumbovi imaju funkcionalnost da, prilikom pritiska, odvedu na odgovarajuća mjesta u aplikaciji.

2.2.6.1. Atributi klase 6

private static Button bookingsButton, customersButton, flightsButton, planesButton, exitButton – atributi tipa Button koji inicijaliziraju gumbove „Bookings“, „Customers“, „Flights“, „Planes“, „Exit“

2.2.7.2. Metode klase 6

public static void initialize() – inicijalizacija glavnog izbornika, postavljanje gumbova i njihove funkcionalnosti

public static void handleBookingsButton() – metoda za prikaz „Bookings“ gumba

public static void handleCustomersButton() – metoda za prikaz „Customers“ gumba

public static void handleFlightsButton() – metoda za prikaz „Flights“ gumba

public static void handlePlanesButton() – metoda za prikaz „Planes“ gumba

public static void handleExitButton() – metoda za prikaz „Exit“ gumba

2.2.7 Klasa 7 – PlaneEditSceneControl

Ova klasa služi za inicijalizaciju i funkcionalnost pomoćnog izbornika o detaljima aviona prilikom dodavanja i uređivanja aviona. Sastoji se od 4 polja za unos teksta, naslova i dva gumba.

2.2.7.1. Atributi klase 7

private static Button okButton, cancelButton – atributi tipa Button koji služe za inicijalizaciju gumbova „Ok“ i „Cancel“

private static TextField planeName, firstClassText, coachText, economyText – atributi za unos teksta/broja sjedala za „First“, „Coach“ i „Economy“ klasu

private static Plane plane – atribut plane/poziv klase Plane koja služi za prikaz/uređivanje već dodanih aviona

private static boolean okPressed = false – boolean atribut kojim pratimo ako je gumb „OK“ pritisnut odnosno ako nije

2.2.7.2. Metode klase 7

public static void initialize() – inicijalizacija prozora za unos/uređivanje letova

public static void setPlane(Plane p) – metoda za dohvat podataka iz baze „Plane“. Koristi se za ispis aviona na glavni ekran i za uređivanje

public static void handleOkButton() funkcionalnost gumba „Ok“, ako je formular popunjen avion se dodaje u bazu, inače javlja grešku

public static void handleClose() – metoda koja zatvara pomoćni prozor

public static boolean isValid() – metoda za provjeru korisničkog unosa. Vraća true ili false

public static Plane getPlane() – getter metoda koja vraća avion

public static boolean isOkPressed() – metoda koja vraća je li gumb „Ok“ pritisnut. Vraća true ili false

2.2.8 Klasa 8 – ViewBookingSceneControl

Klasa za inicijalizaciju prozora za rezervaciju leta. Implementirane su CRUD operacije, moguće je dodati, urediti i izbrisati rezervaciju leta. Također se računa povrat novca prilikom otkaza rezervacije

2.2.8.1. Atributi klase 8

private static TableView<BookingTable> table – atribut tipa TableView koji je dizajniran za vizualizaciju reda podataka za rezervaciju letova, podjeljenih u stupce

private static Label departureDateObs, arrivalDateObs, categoryObs, firstNameObs, lasnNameObs – atributi tipa Label za prikaz teksta „Departure“, „Arrival“, „Category“, „First name“, „Last name“, „Price“

private static Label, priceObs, ageObs, passportObs, phoneNumberObs - atributi tipa Label za prikaz teksta „Price“, „Age“, „Passport“, „Phone number“

private static TextField search – atribut tipa TextField koji služi za pretraživanje rezervacija

private static ObservableList<BookingTable> bookings, tableItems – atribut koji prati promjene ako se dogode

private static double refund – atribut tipa double koji računa povrat novca

2.2.8.2. Metode klase 8

public static void initialize() – metoda za inicijalizaciju „Booking“ prozora, postavlja svih oznaka teksta, search bar, tablicu za unos podataka i rezervaciju letova

public static void handle_backButton() – metoda za postavljanje „Back“ gumba

public static void handle_addButton() – metoda za postavljanje i funkcionalnost „Add“ gumba. Također je definirana funkcionalnost za dodavanje rezervacije u bazu podataka i postavljanje stavki tablice

public static void handle_editButton() – metoda za inicijalizaciju „Edit“ gumba. Implementirana je funkcionalnost gumba, može se promijeniti kategorija karte i uređivanje kupca

public static void handle_cancelButton() – inicijalizacija gumba „Cancel“. Postavljeno je tako da se rezervacija ne može otkazati dva dana prije leta i postavlja se iznos povrata novca

public static void cancelBooking(BookingTable booking table, Booking booking) – metoda za otkazivanje rezervacije. Nakon otkazivanja rezervacije pojavljuje se sporedni ekran koji prikazuje potvrdu da je rezervacija otkazana i prikazuje količinu povrata nova

public static void displayBookingInfo(BookingTable buk) – metoda za prikaz detalja rezervacije

public static void initializeSearch() – inicijalizacija trake za pretraživanje

2.2.9 Klasa 9 – ViewCustomerSceneControl

Klasa za inicijalizaciju prozora za dodavanje kupca. Implementirane su CRUD operacije, moguće je dodati, urediti i izbrisati kupca.

2.2.9.1. Atributi klase 9

public static TableView<Customer> table - atribut tipa TableView koji je dizajniran za vizualizaciju reda podataka, podjeljenih u stupce
private static TextField search – atribut tipa TextField koji služi za pretraživanje kupaca
private static ObservableList<Customer> customers – atribut koji prati promjene ako se dogode
private static ObservableList<Customer> tableItems – atribut koji prati promjene ako se dogode
private static Button backButton, addButton, editButton -atributi tipa Button koji inicijaliziraju gumbove „Back“, „Add“, „Edit“

2.2.9.2. Metode klase 9

public static void initialize() - metoda za inicijalizaciju „Customer“ prozora, postavlja svih oznaka teksta, search bar, tablicu za unos podataka i unos kupaca

public static void handle_addB() - metoda za postavljanje i funkcionalnost „Add“ gumba. Također je definirana funkcionalnost za dodavanje kupca u bazu podataka i postavljanje stavki tablice

public static void handle_editB() – metoda za inicijalizaciju „Edit“ gumba. Implementirana je funkcionalnost gumba, može se promijeniti ime, prezime, dob kupca, broj telefona i putovnice

public static void initializeSearch() – inicijalizacija trake za pretraživanje

public static void handle_backB() – metoda za postavljanje „Back“ gumba

2.2.10 Klasa 10 – ViewFlightSceneControl

Klasa za inicijalizaciju prozora za dodavanje letova. Implementirane su CRUD operacije, moguće je dodati, urediti i izbrisati let.

2.2.10.1. Atributi klase 10

private static TableView<FlightTable> table - atribut tipa TableView koji je dizajniran za vizualizaciju reda podataka, podjeljenih u stupce
private static ObservableList<FlightTable> flights, tableItems – atribut koji prati promjene ako se dogode
private static TextField search– atribut tipa TextField koji služi za pretraživanje leta
private static Button backButton, addButton, editButton, exportButton – atribut tipa Button za dodavanje gumbova „Back“, „Add“, „Edit“, „Export“

2.2.10.2. Metode klase 10

public static void initialize() – metoda za inicijalizaciju „Flights“ prozora, postavlja svih oznaka teksta, search bar, tablicu za unos podataka i unos letova

public static void handle_addB()- metoda za postavljanje i funkcionalnost „Add“ gumba. Također je definirana funkcionalnost za dodavanje leta u bazu podataka i postavljanje stavki tablice

public static void handle_editB() – metoda za inicijalizaciju „Edit“ gumba. Implementirana je funkcionalnost gumba, može se promijeniti ruta, datum, vrijeme polaska, ID aviona i početna cijena

public static void handle_backB() – metoda za postavljanje „Back“ gumba

public static void handle_exportFlightsB() – metoda za postavljanje gumba za izvoz letova u .txt datoteku

public static void initializeSearch() – inicijalizacija trake za pretraživanje

2.2.11 Klasa 11 – ViewPlaneSceneControl

Klasa za inicijalizaciju prozora za dodavanje aviona. Implementirane su CRUD operacije, moguće je dodati, urediti i izbrisati avion.

2.2.11.1. Atributi klase 11

private static TableView<Plane> table - atribut tipa TableView koji je dizajniran za vizualizaciju reda podataka, podjeljenih u stupce

private static Button addButton, backButton, editButton, exportPlaneButton – atribut tipa Button za dodavanje gumbova „Back“, „Add“, „Edit“, „Export“

2.2.11.2. Metode klase 11

public static void initialize() – metoda za inicijalizaciju „Planes“ prozora, postavlja svih oznaka teksta, search bar, tablicu za unos podataka i unos letova

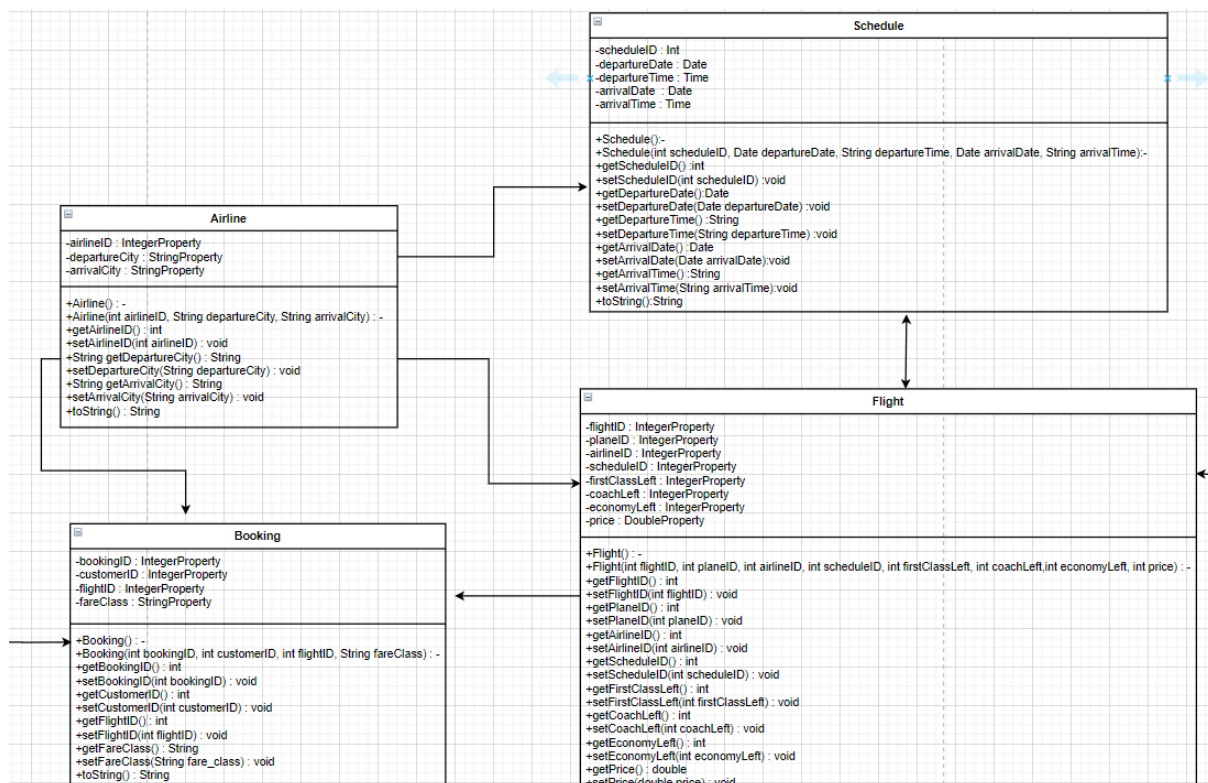
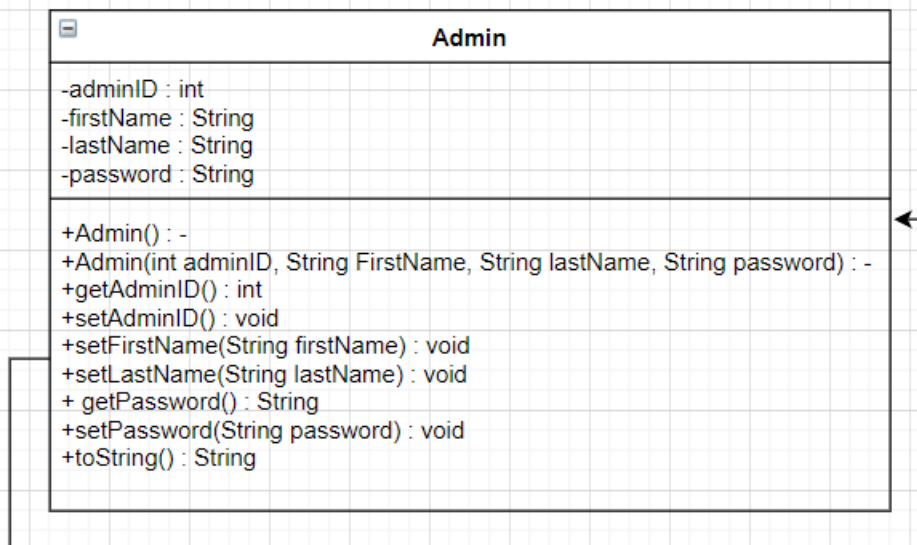
public static void handle_addB() – metoda za postavljanje i funkcionalnost „Add“ gumba. Također je definirana funkcionalnost za dodavanje aviona u bazu podataka i postavljanje stavki tablice

public static void handle_editB() – metoda za inicijalizaciju „Edit“ gumba. Implementirana je funkcionalnost gumba, može se promijeniti ime aviona, broj mjesta u First, Coach i Economy klasi

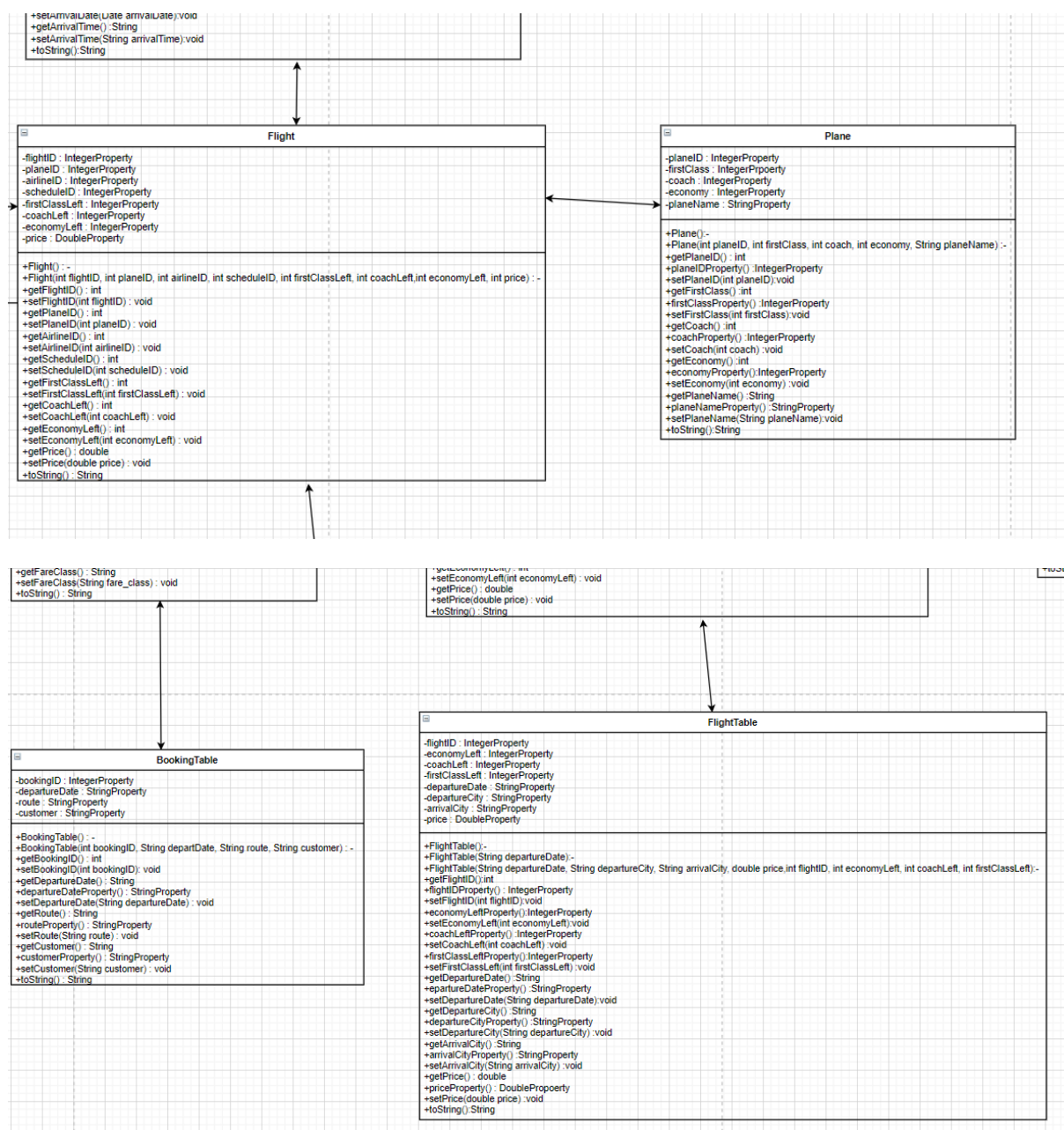
public static void handle_backB() – metoda za postavljanje „Back“ gumba

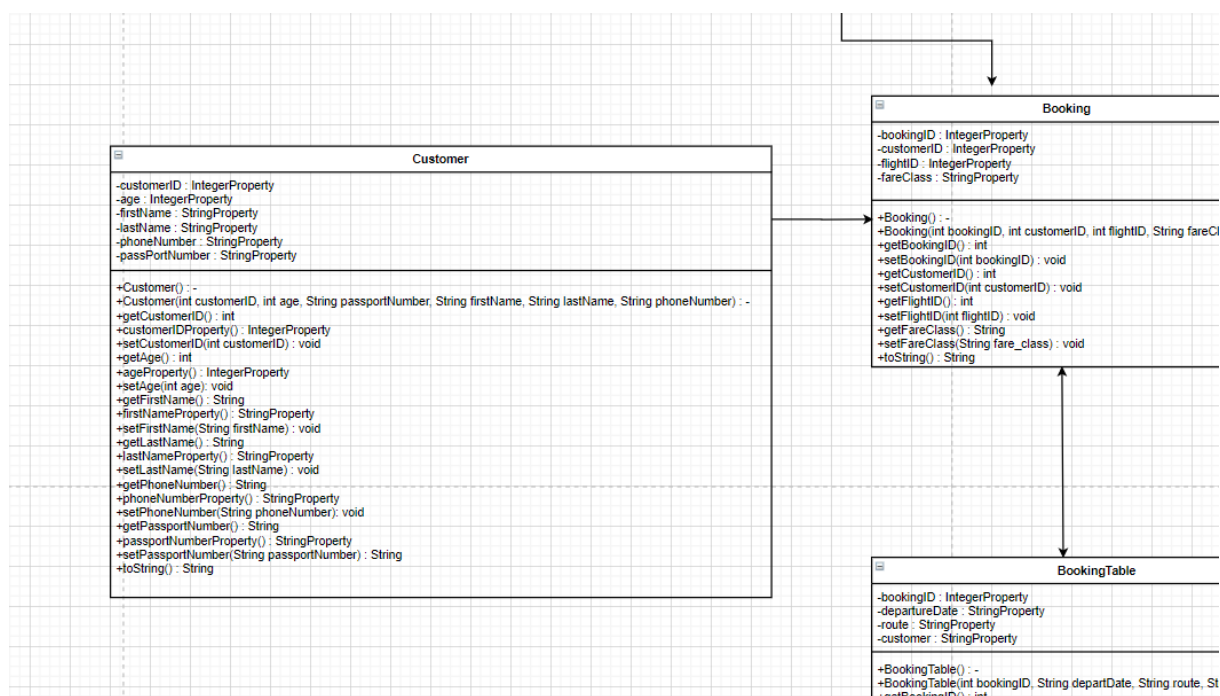
public static void handle_exportFlightsB() – metoda za postavljanje gumba za izvoz aviona u .txt datoteku

2.3. Klase - Application: DataTypes



Tehnička dokumentacija projekta





2.3.1. Klasa 1 - Admin

Klasa koja realizira Admina.

2.3.1.1. Atributi

private int adminID – identifikacijski broj admina

private String firstName – ime admina

private String lastName - prezime admina

private String password – šifra admina

2.3.1.2. Metode

Public admin() – defaultni konstruktor

Public admin(int adminID, String firstName, String lastName, String password) - konstruktor sa svim parametrima

public int getAdminID() – getter metoda za adminID

public void setAdminID(int adminID) – setter metoda za adminID

public void setFirstName(String firstName) – setter metoda za firstName

public void setLastName(String lastName) – setter metoda lastName

public String getPassword() – getter metoda za password

public void setPassword(String password) – setter metoda za password

public String toString() – Override metoda toString koja ispisuje atribut admina

2.3.2. Klasa 2 - Airline

Klasa koja definira zrakoplovnu kompaniju i rute koje pruža.

2.3.2.1. Atributi

private IntegerProperty airlineID – identifikacijski broj avionske kompanije
private StringProperty departureCity, arrivalCity – grad polaska i dolaska

2.3.2.2. Metode

public Airline() – defaultni konstruktor

public Airline(int airlineID, String departureCity, String arrivalCity) – konstruktor sa svim parametrima

public int getAirlineID() – getter za airlineID

public void setAirlineID(int airlineID) – setter za airlineID

public String getDepartureCity() – getter za departureCity

public void setDepartureCity(String departureCity) – setter za departureCity

public String getArrivalCity() – getter za arrivalCity

public void setArrivalCity(String arrivalCity) – setter za arrivalCity

public String toString() – override metoda toString koja ispisuje atribute airline

2.3.3. Klasa 3 - Booking

Klasa koja definira rezervaciju leta

2.3.3.1. Atributi

private IntegerProperty bookingID, customerID, flightID – atributi za identifikacijski broj leta, kupca i leta

private String fareClass – atribut za odabir klase leta

2.3.3.2. Metode

public Booking() – defaultni konstruktor

public Booking(int bookingID, int customerID, int flightID, String fareClass) – konstruktor sa svim parametrima

public int getBookingID() – getter za bookingID

public void setBookingID(int bookingID) – setter za bookingID

public int getCustomerID() – getter za customerID

public void setCustomerID(int customerID) – setter za customerID

public int getFlightID() - getter za flightID

public void setFlightID(int flightID) - setter za flightID

public String getFareClass() - getter za fareClass

public void setFareClass(String fare_class) - setter za fareClass

public String toString() – override metoda toString koja ispisuje atribute booking

2.3.4. Klasa 4 - BookingTable

Ova klasa definira „povezivanje“ kupca s rezervacijom.

2.3.4.1. Atributi

private IntegerProperty bookingID – atribut za identifikacijski broj rezervacije

private StringProperty departureDate, route, customer – atributi za odabir datuma polaska, rute i kupca

2.3.4.2. Metode

public Booking() – defaultni konstruktor

public Booking(int bookingID, String departureDate, String route, String customer) – konstruktor sa svim parametrima

public int getBookingID() – getter za identifikaciju rezervacije

public void setBookingID(int bookingID) – setter za identifikaciju rezervacije

public String getDepartureDate() – getter za datum polaska

public StringProperty departureDateProperty() – getter za datum polaska

public void setDepartureDate(String departureDate) – setter za datum polaska

public String getRoute() – getter za rutu

public StringProperty routeProperty() – getter za rutu

public void setRoute(String route) – setter za rutu

public String getCustomer() – getter za kupca

public StringProperty customerProperty() – getter za kupca

public void setCustomer(String customer) – setter za kupca

public String toString() – override metoda toString koja ispisuje atribute BookingTble

2.3.5. Klasa 5 - Customer

Klasa koja kreira kupca.

2.3.5.1. Atributi

private IntegerProperty customerID, age – atributi za identifikacijski broj kupca i njegovu dob

private StringProperty firstName, lastName, phoneNumber, passportNumber – atributi za ime, prezime, broj mobitela i putovnice kupca

2.3.5.2. Metode

public Customer() – konstruktor bez parametara

public Customer(int customerID, int age, String passportNumber, String firstName, String lastName, String phoneNumber) – konstruktor sa svim parametrima

public int getCustomerID() – getter za identifikaciju kupca

public IntegerProperty customerIDProperty() – getter za identifikaciju kupca

public void setCustomerID(int customerID) – setter za identifikaciju kupca

public int getAge() – getter za dob kupca

public IntegerProperty ageProperty() – getter za dob kupca

public void setAge(int age) – setter za dob kupca

public String getFirstName() – getter za ime kupca

public StringProperty firstNameProperty() – getter za ime kupca

public void setFirstName(String firstName) – setter za ime kupca

public String getLastName() – getter za prezime kupca

public StringProperty lastNameProperty() – getter za prezime kupca

public void setLastName(String lastName) – setter za prezime kupca

public String getPhoneNumber() – getter za broj telefona kupca

public StringProperty phoneNumberProperty() – getter za broj telefona kupca

public void setPhoneNumber(String phoneNumber) – setter za broj telefona kupca

public String getPassportNumber() – getter za broj putovnice kupca

public StringProperty passportNumberProperty() – getter za broj putovnice kupca

public void setPassportNumber(String passportNumber) – setter za broj putovnice kupca

public String toString() – override metoda toString koja ispisuje atribute customer

2.3.6. Klasa 6 - Flight

Ova klasa kreira let kojim kupac ide do odredišta.

2.3.6.1. Atributi

private IntegerProperty flightID, planeID, airlineID, scheduleID, firstClassLeft, coachLeft, economyLeft – atributi za identifikacijske brojeve leta, aviona, avionske kompanije, rasporeda, koliko je sjedišta ostalo u First, Coach i Economy klasi

private DoubleProperty price – atribut kojim se kreira cijena leta

2.3.6.2. Metode

public Flight() – default konstruktor

public Flight(int flightID, int planeID, int airlineID, int scheduleID, int firstClassLeft, int coachLeft, int economyLeft, int price) – konstruktro sa svim parametrima

public int getFlightID() – getter za flightID

public void setFlightID(int flightID) – setter za flightID

public int getPlaneID() – getter za planeID

public void setPlaneID(int planeID) – setter za planeID

public int getAirlineID() – getter za airlineID

public void setAirlineID(int airlineID) – setter za airlineID

public int getScheduleID() – getter za scheduleID

public void setScheduleID(int scheduleID) – setter za scheduleID

public int getFirstClassLeft() – getter za firstClassLeft

public void setFirstClassLeft(int firstClassLeft) – setter za firstClassLeft

public int getCoachLeft() – getter za coachLeft

public void setCoachLeft(int coachLeft) – setter za coachLeft

public int getEconomyLeft() – getter za economyLeft

public void setEconomyLeft(int economyLeft) – setter za economyLeft

public double getPrice() – getter za price

public void setPrice(double price) – setter za price

public String toString() – override metoda toString koja ispisuje atribut flight

2.3.7. Klasa 7 - FlightTable

Ova klasa definira „povezivanje“ let s rasporedom i zrakoplovnom kompanijom.

2.3.7.1. Atributi

private IntegerProperty flightID, economyLeft, coachLeft, firstClassLeft – atributi za kreiranje identifikacijskog broja leta, ostatka sjedišta u Economy, Coach i First Class klasama
private StringProperty departureDate, departureCity, arrivalCity – atributi za kreiranje datuma polaska, grada polaska i dolaska
private DoubleProperty price – atribut za kreiranje cijene

2.3.7.2. Metode

public FlightTable() -default konstruktor

public FlightTable(String departureDate) konstruktor s jednim parametrom

public FlightTable(String departureDate, String departureCity, String arrivalCity, double price, int flightID, int economyLeft, int coachLeft, int firstClassLeft) – konstruktor sa svim parametrima

public int getFlightID() - getter za identifikacijski broj leta

public IntegerProperty flightIDProperty() - getter za identifikacijski broj leta

public void setFlightID(int flightID) - setter za identifikacijski broj leta

public IntegerProperty economyLeftProperty() - getter za economyLeft

public void setEconomyLeft(int economyLeft) - setter za economyLeft

public IntegerProperty coachLeftProperty() - getter za coachLeft

public void setCoachLeft(int coachLeft) - setter za coachLeft

public IntegerProperty firstClassLeftProperty() - getter za firstClassLeft

public void setFirstClassLeft(int firstClassLeft) - setter za firstClassLeft

public String getDepartureDate() - getter za departureDate

public StringProperty departureDateProperty() - getter za departureDate

public void setDepartureDate(String departureDate) - setter za departureDate

public String getDepartureCity() - getter za departureCity

public StringProperty departureCityProperty() - getter za departureCity
public void setDepartureCity(String departureCity) - setter za departureCity
public String getArrivalCity() - getter za arrivalCity
public StringProperty arrivalCityProperty() - getter za arrivalCity
public void setArrivalCity(String arrivalCity) - setter za arrivalCity
public double getPrice() - getter za price
public DoubleProperty priceProperty() - getter za price
public void setPrice(double price) - setter za price
public String toString() – override metoda toString koja ispisuje atribute FlightTable

2.3.8. Klasa 8 - Plane

Ova klasa konstruira zrakoplov kojim se ide od mjesta polaska do odredišta.

2.3.8.1. Atributi

private IntegerProperty planeID, firstClass, coach, economy -atributi za kreiranje identifikacijskog broja zrakoplova, broja sjedišta First, Coach i Economy klasa
private StringProperty planeName – atribut za kreiranje imena zrakoplova

2.3.8.2. Metode

public Plane() – default konstruktor
public Plane(int planeID, int firstClass, int coach, int economy, String planeName) – konstruktor sa svim parametrima
public int getPlaneID() - getter za planeID
public IntegerProperty planeIDProperty() - getter za planeID
public void setPlaneID(int planeID) - setter za planeID
public int getFirstClass() - getter za firstClass
public IntegerProperty firstClassProperty() - getter za firstClass
public void setFirstClass(int firstClass) - setter za firstClass
public int getCoach() - getter za coach
public IntegerProperty coachProperty() - getter za coach
public void setCoach(int coach) - setter za coach

public int getEconomy() - getter za economy
public IntegerProperty economyProperty() - getter za economy
public void setEconomy(int economy) - setter za economy
public String getPlaneName() - getter za planeName
public StringProperty planeNameProperty() - getter za planeName
public void setPlaneName(String planeName) - setter za planeName
public String toString() – override metoda toString koja ispisuje atribute Plane

2.3.9. Klasa 9 - Schedule

Ova klasa kreira raspored kojim se avionska kompanija služi za kreiranje letova

2.3.9.1. Atributi

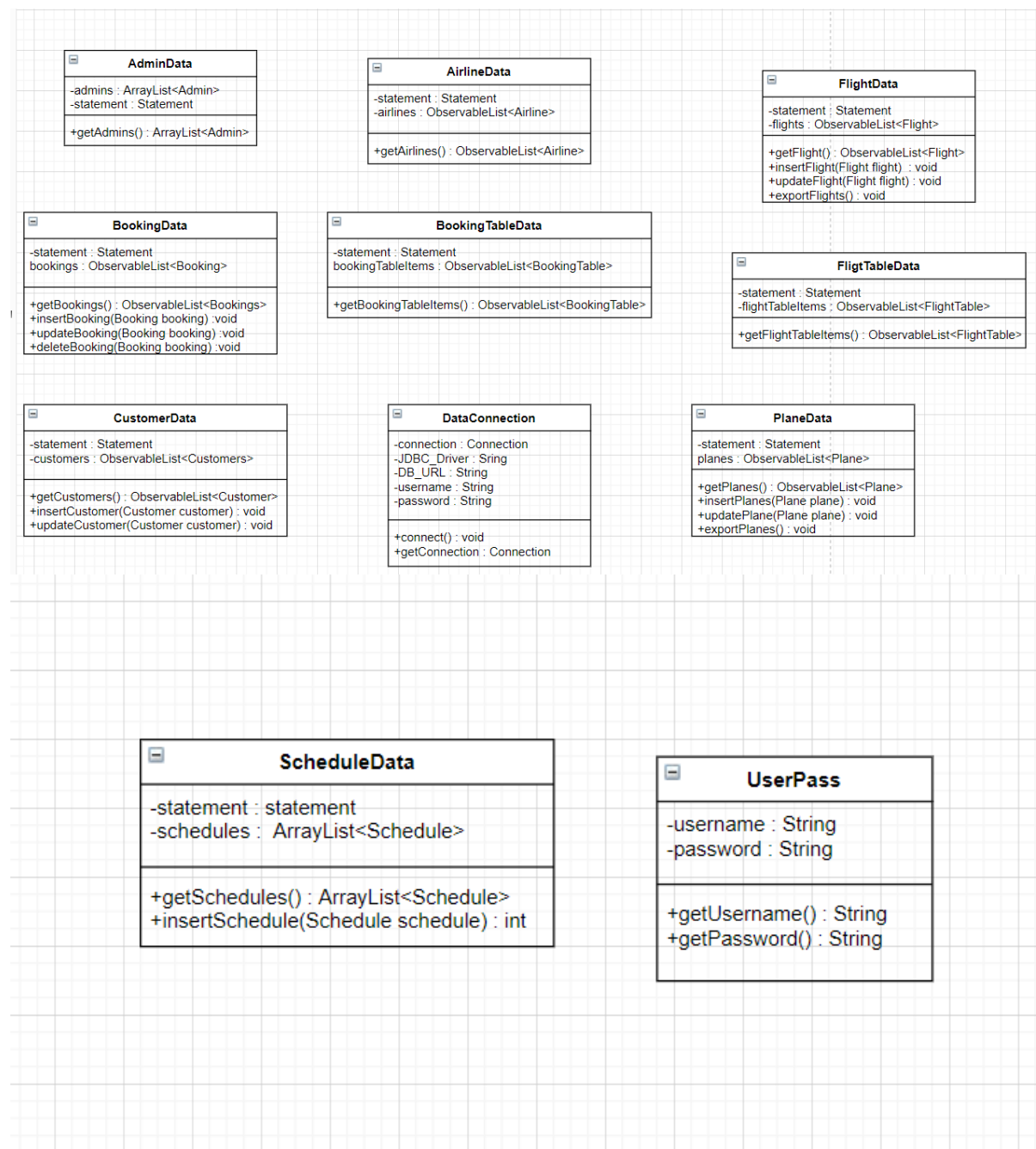
private int scheduleID - atribut za kreiranje identifikacijskog broja rasporeda
private Date departureDate – atribut za kreiranje datuma polaska
private String departureTime - atribut za kreiranje satnice polaska
private Date arrivalDate – atribut za kreiranje datuma dolaska
private String arrivalTime – atribut za kreiranje satnice dolaska

2.3.9.2. Metode

public Schedule() – default konstruktor
public Schedule(int scheduleID, Date departureDate, String departureTime, Date arrivalDate, String arrivalTime) – konstruktor sa svim parametrima
public int getScheduleID() – getter metoda za scheduleID
public void setScheduleID(int scheduleID) – setter metoda za scheduleID
public Date getDepartureDate() – getter metoda za departureDate
public void setDepartureDate(Date departureDate) – setter metoda za departureDate
public String getDepartureTime() – getter metoda za departureTime
public void setDepartureTime(String departureTime) – setter metoda za departureTime
public Date getArrivalDate() – getter metoda za arrivalDate
public void setArrivalDate(Date arrivalDate) – setter metoda za arrivalDate
public String getArrivalTime() – getter metoda za arrivalTime
public void setArrivalTime(String arrivalTime) – setter metoda za arrivalTime

public String toString()– override metoda toString koja ispisuje atribute schedule

2.4. Klase - DataAccess



2.4.1. Klasa 1 - AdminData

Klasa AdminData dohvaća informacije o adminima iz baze.

2.4.1.1. Atributi

private static ArrayList<Admin> admins – atribut tipa ArrayList u kojem se spremaju podaci o adminima

private static Statement statement -atribut tipa Statement uz pomoć kojega ćemo izvršavati SQL query-je

2.4.1.2. Metode

public static ArrayList<Admin> getAdmins() – metoda kojom iz baze dohvaćamo podatke o zrakoplovnim kompanijama i spremamo ih u listu

2.4.2. Klasa 2 - AirlineData

Klasa AdminData dohvaća informacije o adminima iz baze.

2.4.2.1. Atributi

private static Statement statement -atribut tipa Statement uz pomoć kojega ćemo izvršavati SQL query-je

private static ObservableList<Airline> airlines – atribut tipa ObservableList u kojem se spremaju podaci o zrakoplovnim kompanijama i rutama

2.4.2.2. Metode

public static ObservableList<Airline> getAirlines() – metoda kojom iz baze dohvaćamo podatke o zrakoplovnim kompanijama i spremamo ih u listu .

2.4.3. Klasa 3 - BookingData

Klasa BookingData dohvaća informacije o rezervacijama iz baze.

2.4.3.1. Atributi

private static Statement statement -atribut tipa Statement uz pomoć kojega ćemo izvršavati SQL query-je

private static ObservableList<Booking> bookings – atribut tipa ObservableList u kojem se spremaju podaci o rezervacijama letova

2.4.3.2. Metode

public static ObservableList<Airline> getAirlines() – metoda kojom iz baze dohvaćamo podatke o rezervacijama letova i spremamo ih u listu .

2.4.4. Klasa 4 - BookingTableData

Ova klasa dohvaća podatke iz baze, povezivajući 5 različitih tablica (booking, flight, customer, schedule, airline), pomažući nam kod rezervacije leta.

2.4.4.1. Atributi

private static Statement statement -atribut tipa Statement uz pomoć kojega ćemo izvršavati SQL query-je

private static ObservableList<Booking> bookings – atribut tipa ObservableList u kojem se spremaju podaci o rezervacijama letova

2.4.4.2. Metode

public static ObservableList<BookingTable> getBookingTableItems() – metoda koja čita iz baze i pritom povezuje tablice.

2.4.5. Klasa 1 - CustomerData

Klasa dohvaća podatke o kupcima iz baze, također vrši unos i ažuriranje kupaca.

2.4.5.1. Atributi

private static Statement statement -atribut tipa Statement uz pomoć kojega ćemo izvršavati SQL query-je

private static ObservableList<Customer> customers – atribut tipa ObservableList u kojem se spremaju podaci o rezervacijama kupcima

2.4.5.2. Metode

public static ObservableList<Customer> getCustomers() – metoda koja čita iz baze i dohvaća podatke o kupcima

public static void insertCustomer(Customer customer) – metoda koja vrši unos podataka o kupcima

public static void updateCustomer(Customer customer) - metoda koja vrši ažuriranje podataka kupca

2.4.6. Klasa 6 - DataConnection

Klasa u kojoj se vrši spajanje na bazu.

2.4.6.1. Atributi

private static Connection connection – atribut pomoću kojeg se spajamo na bazu

private static String JDBC_Driver = " " – komponenta pomoću kojag program komunicira s bazom

private static final String DB_URL = " " – atribut koji sadži link baze

private static String username = UserPass.getUsername() – atribut koji dohvaća username baze

private static String password = UserPass.getPassword() – atribut koji dohvaća password baze

2.4.6.2. Metode

public static void connect() – metoda koja vrši spajanje na bazu

public static Connection getConnection() - getter

2.4.7. Klasa 7 - FlightData

Klasa dohvaća podatke o kupcima iz baze, također vrši unos i ažuriranje letova, te izvoz letova u .txt datoteku.

2.4.7.1. Atributi

private static Statement statement -atribut tipa Statement uz pomoć kojega ćemo izvršavati SQL query-je

private static ObservableList<Flight> flights – atribut tipa ObservableList u kojem se spremaju podaci o rezervacijama letovima

2.4.7.2. Metode

public static ObservableList<Flight> getFlight() – metoda koja dohvaća elemente iz baze

public static void insertFlight(Flight flight) - metoda za dodavanje leta

public static void updateFlight(Flight flight) - metoda za ažuriranje leta

public static void exportFlights() – metoda za izvoz podataka o letu u .txt datoteku

2.4.8. Klasa 8 - FlightTableData

Ova klasa dohvaća podatke iz baze, povezivajući 3 različitih tablica (flight, schedule, airline), pomažući nam kod rezervacije leta.

2.4.8.1. Atributi

private static Statement statement -atribut tipa Statement uz pomoć kojega ćemo izvršavati SQL query-je

private static ObservableList<FlightTable> flightTableItems – atribut tipa ObservableList u kojem se spremaju podaci o letovima, raspredu i zrakoplovnoj kompaniji

2.4.8.2. Metode

public static ObservableList<FlightTable> getFlightTableItems() – metoda koja spaja 3 baze i dohvaća podatke iz njih

2.4.9. Klasa 9 - PlaneData

Ova klasa dohvaća podatke iz tablice plane te implementirane su metode za unos i ažuriranje podataka te izvoz podataka u .txt datoteku

2.4.9.1. Atributi

private static Statement statement -atribut tipa Statement uz pomoć kojega ćemo izvršavati SQL query-je

private static ObservableList<Plane> planes – atribut tipa ObservableList u kojem se spremaju podaci o zrakoplovima

2.4.9.2. Metode

public static ObservableList<Plane> getPlanes() – metoda koja dohvaća podatke o zrakoplovima

public static void insertPlanes(Plane plane) – metoda koja unosi podatke u bazu plane

public static void updatePlane(Plane plane) – metoda koja ažurira podatke o zrakoplovima

public static void exportPlanes() – metoda koja izvodzi podatke u .txt datoteku

2.4.10. Klasa 10 - ScheduleData

Klasa koja definira raspored kojim se zrakoplovna kompanija vodi za rezerviranje letova i unos podataka u bazu

2.4.10.1. Atributi

private static Statement statement -atribut tipa Statement uz pomoć kojega ćemo izvršavati SQL query-je

private static ObservableList<Schedule> schedules – atribut tipa ObservableList u kojem se spremaju podaci o rasporedima

2.4.10.2. Metode

public static ArrayList<Schedule> getSchedules() – metoda koja dohvaća podatke o rasporedu iz baze

public static int insertSchedule(Schedule schedule) – metoda koja unosi podatke u tablic schedule

2.4.11. Klasa 11 - UserPass

Klasa koja definira korisničko ime i lozinku baze

2.4.11.1. Atributi

private static String username – atribut kojim se definira korisničko ime

private static String password– atribut kojim se definira korisničko lozinka

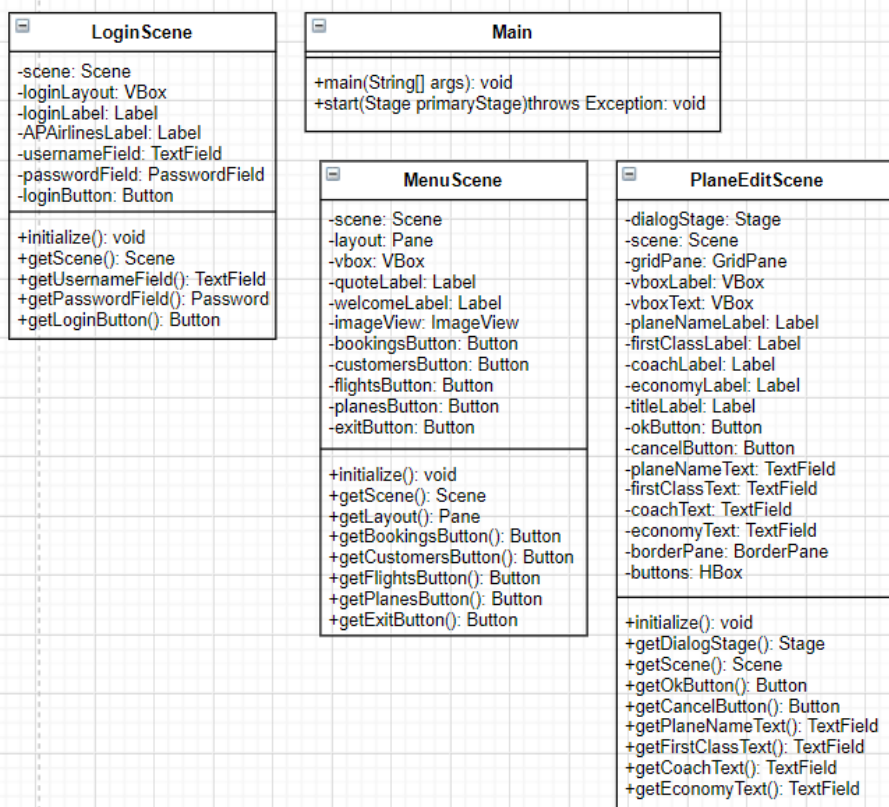
2.4.11.2. Metode

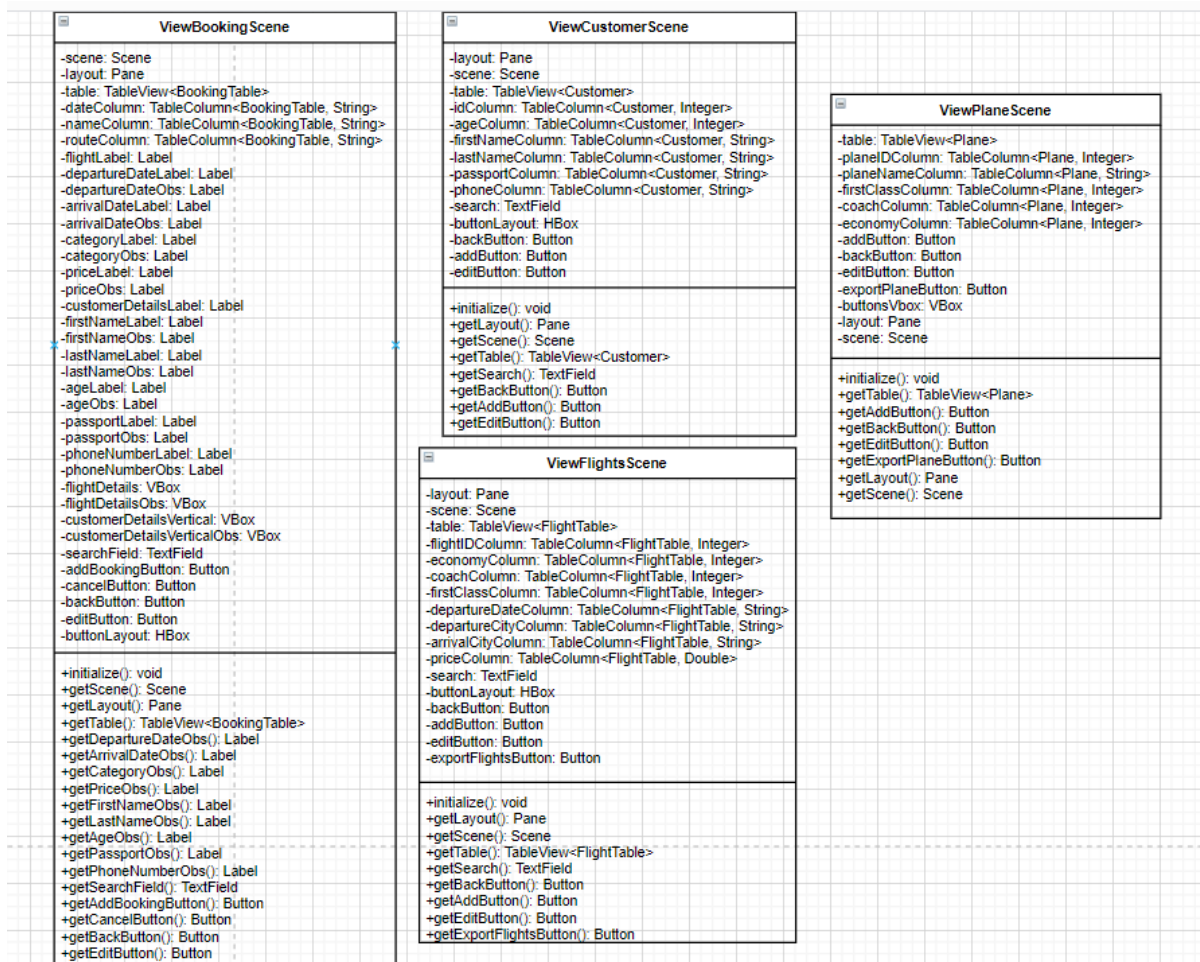
public static String getUsername() – metoda(getter) koja dohvaća korisničko ime

public static String getPassword()– metoda(getter) koja dohvaća korisničko lozinku

2.5. Klase - sample

BookingEditScene	CustomerEditScene	FlightsEditScene
-dialogStage : Stage -scene : Scene -layout : Pane -customerLabel: Label -addCustomerLabel: Label -bookingLabel: Label -routeLabel: Label -departureDateLabel: Label -categoryLabel: Label -priceLabel: Label -labelsLayout: VBox -detailsLayout: VBox -customerBox: ComboBox<String> -routeBox: ComboBox<String> -departureDatePicker: DatePicker -firstClass: RadioButton -coachClass: RadioButton -economyClass: RadioButton -g: ToggleGroup -toggleBox: HBox -addCustomer: Button -okButton: Button -cancelButton: Button -priceLabelObs: Label -buttons: HBox +initialize(): void +getDialogStage(): Stage +getScene(): Scene +getLayout(): Pane +getCustomerBox(): ComboBox<String> +getRouteBox(): ComboBox<String> +getDepartureDatePicker(): DatePicker +getFirstClass(): RadioButton +getCoachClass(): RadioButton +getEconomyClass(): RadioButton +getAddCustomer(): Button +getOkButton(): Button +getCancelButton(): Button +getPriceLabelObs(): Label	-dialogStage: Stage -scene: Scene -gridPane: GridPane -vBoxLabel: VBox -vBoxText: VBox -firstName: Label -lastName: Label -age: Label -phoneNumber: Label -passportNumber: Label -title: Label -okButton: Button -cancelButton: Button -firstNameText: TextField -lastNameText: TextField -ageText: TextField -phoneNumberText: TextField -passportNumberText: TextField -borderPane: BorderPane -buttons: HBox +initialize(): void +getDialogStage(): Stage +getScene(): Scene +getOkButton(): Button +getCancelButton(): Button +getFirstNameText(): TextField +getLastNameText(): TextField +getAgeText(): TextField +getPhoneNumberText(): TextField +getPassportNumberText(): TextField	-dialogStage: Stage -scene: Stage -layout: Pane -flightLabel: Label -timeLabel: Label -routeLabel: Label -dateLabel: Label -planeLabel: Label -toLabel: Label -priceLabel: Label -route: ComboBox<String> -departureTime: ComboBox<String> -arrivalTime: ComboBox<String> -planeID: ComboBox<Integer> -departureDate: DatePicker -price: TextField -okButton: Button -cancelButton: Button -vBox1: VBox -vBox2: VBox -timeBox: HBox -buttons: HBox +initialize(): void +getDialogStage(): Stage +getScene(): Scene +getLayout(): Pane +getRoute(): ComboBox<String> +getDepartureTime(): ComboBox<String> +getArrivalTime(): ComboBox<String> +getPlaneID(): ComboBox<Integer> +getDepartureDate(): DatePicker +getPrice(): TextField +getOkButton(): Button +getCancelButton(): Button





2.5.1. Klasa 1 - BookingEditScene

Klasa „BookingEditScene“ inicijalizira frontend dio pomoćnog prozora za rezerviranje leta.

2.5.1.1. Atributi

private static Stage dialogStage – atribut koji sadrži sve objekte aplikacije
private static Scene scene – atribut koji predstavlja fizički sadržaji aplikacije
private static Pane layout – atribut kojim se ručno može rasporediti korisničko sučelje postavljanjem svojstava položaja i veličine za svaki element
private static Label customerLabel, addCustomerLabel, bookingLabel, routeLabel, departureDateLabel – non-editable atributi za „Customer“, „Add customer“, „Booking“, „Route“, „Departure date“
private static Label categoryLabel, priceLabel – non-editable atributi za „Category“, „Price“
private static VBox labelsLayout, detailsLayout – atributi koji postavljaju elemente u vertikalni stupac
private static ComboBox<String> customerBox – atribut koji omogućava prikazivanje listu kupaca koje možemo odabrati
private static ComboBox<String> routeBox – atribut koji omogućava prikazivanje listu ruta koje možemo odabrati
private static DatePicker departureDatePicker – atribut za odabir datuma polaska
private static RadioButton firstClass – atribut za odabir First class

private static RadioButton coachClass – atribut za odabir Coach class
private static RadioButton economyClass – atribut za odabir Economy class
private static ToggleGroup g – atribut za odabir klase sjedenja
private static HBox toggleBox – atribut koji postavlja elemente u vodoravni red
private static Button addCustomer – atribut za dodavanje gumba „Add customer“
private static Button okButton – atribut za dodavanje gumba „Ok“
private static Button cancelButton – atribut za dodavanje gumba „Close“
private static Label priceLabelObs - non-editable atribut za „Price“
private static HBox buttons – atribut koji postavlja elemente(gumbove) u vodoravni red

2.5.1.2. Metode

public static void initialize() – metoda za inicijalizaciju frontend dijela pomoćnog prozora za rezervaciju leta

public static Stage getDialogStage() – getter metoda za „dialogStage“

public static Scene getScene() – getter metoda za „scene“

public static Pane getLayout() - getter metoda za „layout“

public static ComboBox<String> getCustomerBox() – getter metoda za „customerBox“

public static ComboBox<String> getRouteBox() -getter metoda za „routeBox“

public static DatePicker getDepartureDatePicker() – getter metoda za „departureDatePicker“

public static RadioButton getFirstClass() – getter metoda za „firstClass“ gumb

public static RadioButton getCoachClass() – getter metoda za „coachClass“ gumb

public static RadioButton getEconomyClass() – getter metoda za „economyClass“ gumb

public static Button getAddCustomer()– getter metoda za „addCustomers“ gumb

public static Button getOkButton() – getter metoda za „Ok“ gumb

public static Button getCancelButton() – getter metoda za „Cancel“ gumb

public static Label getPriceLabelObs()– getter metoda za „priceLabelObs“

2.5.2. Klasa 2 - CustomerEditScene

Klasa „CustomerEditScene“ inicijalizira frontend dio pomoćnog prozora za dodavanje i uređivanje kupca.

2.5.2.1. Atributi

private static Stage dialogStage – atribut koji sadrži sve objekte aplikacije

private static Scene scene – atribut koji predstavlja fizički sadržaji aplikacije

private static GridPane gridPane – atribut koji postavlja elemente unutar fleksibilne mreže redaka i stupaca

private static VBox vboxLabel, vboxText – atributi koji postavljaju elemente u vertikalni stupac

private static Label firstName, lastName, age, phoneNumber, passportNumber, title – non-editable atributi za „firstName“, „lastName“, „age“, „phoneNumber“, „passportNumber“, „title“

private static Button okButton, cancelButton – atributi za dodavanje gumbova „Ok“ i „Cancel“

private static TextField firstNameText, lastNameText, ageText, phoneNumberText, passportNumberText – atributi za unos teksta za ime, prezime, dob, broj telefona/mobitela i outovnice

private static BorderPane borderPane – atribut koji postavlja elemente u gornji, lijevi, desni, donji i središnji položaj

private static HBox buttons – atribut koji postavlja elemente(gumbove) u vodoravni red

2.5.2.2. Metode

public static void initialize() – metoda za inicijalizaciju frontend dijela pomoćnog prozora za dodavanje kupca

public static Stage getDialogStage() – getter metoda za „dialogStage“

public static Scene getScene() – getter metoda za „scene“

public static Button getOkButton() – getter metoda za „okButton“

public static Button getCancelButton() – getter metoda za „cancelButton“

public static TextField getFirstNameText() – getter metoda za „firstNameText“

public static TextField getLastNameText() – getter metoda za „lastNameText“

public static TextField getAgeText() – getter metoda za „ageText“

public static TextField getPhoneNumberText() – getter metoda za „phoneNumberText“

public static TextField getPassportNumberText() – getter metoda za „passportNumberText“

2.5.3. Klasa 3 - FlightsEditScene

Klasa „FlightsEditScene“ inicijalizira frontend dio pomoćnog prozora za dodavanje i uređivanje leta.

2.5.3.1. Atributi

private static Stage dialogStage – atribut koji sadrži sve objekte aplikacije

private static Scene scene – atribut koji predstavlja fizički sadržaji aplikacije

private static Pane layout – atribut kojim se ručno može rasporediti korisničko sučelje postavljanjem svojstava položaja i veličine za svaki element

private static Label flightLabel, timeLabel, routeLabel, dateLabel, planeLabel, toLabel, priceLabel – non-editable atributi za „flight“, „time“, „route“, „date“, „plane“, „to“, „price“

private static ComboBox<String> route,departureTime, arrivalTime – atribut koji omogućava prikazivanje listu ruta, vremena polaska i dolaska koje možemo odabrati

private static ComboBox<Integer> planeID – atribut koji omogućava prikazivanje listu identifikacijskog broja leta koje možemo odabrati
private static DatePicker departureDate – atribut za odabir datuma polaska
private static TextField price – atributi za unos teksta za cijenu
private static Button okButton, cancelButton – atributi za dodavanje gumbova „Ok“ i „Cancel“
private static VBox vbox1, vbox2 – atributi koji postavljaju elemente u vertikalni stupac
private static HBox timeBox, buttons – atribut koji postavlja elemente u vodoravni red

2.5.3.2. Metode

public static void initialize() – metoda za inicijalizaciju frontend dijela pomoćnog prozora za dodavanje letova

public static Stage getDialogStage() – getter metoda za „dialogStage“

public static Scene getScene() – getter metoda za „scene“

public static Pane getLayout() - getter metoda za „layout“

public static ComboBox<String> getRoute() - getter metoda za „routeBox“

public static ComboBox<String> getDepartureTime() - getter metoda za „departureTime“

public static ComboBox<String> getArrivalTime() - getter metoda za „arrivalTime“

public static ComboBox<Integer> getPlaneID() - getter metoda za „planeID“

public static DatePicker getDepartureDate() - getter metoda za „departureDate“

public static TextField getPrice() - getter metoda za „price“

public static Button getOkButton() - getter metoda za „okButton“

public static Button getCancelButton() - getter metoda za „cancelButton“

2.5.4. Klasa 4 - LoginScene

Klasa „LoginScene“ inicijalizira frontend dio pomoćnog prozora za logiranje u aplikaciju.

2.5.4.1. Atributi

private static Scene scene – atribut koji predstavlja fizički sadržaji aplikacije
private static VBox loginLayout – atribut koji postavlja element („login“ tekst) u vertikalni stupac
private static Label loginLabel – non-editable atribut za dodavanje „Login“ teksta
private static Label APAirlinesLabel – non-editable atribut za dodavanje „APAAirlines“ naslov teksta
private static TextField usernameField – atribut za unos username teksta
private static PasswordField passwordField – atribut za unos password teksta
private static Button loginButton – atribut za dodavanje gumba „Login“

2.5.4.2. Metode

public static void initialize() – metoda za inicijalizaciju frontend dijela za login ekran

public static Scene getScene() – getter metoda za „scene“

public static TextField getUsernameField() – getter metoda za „usernameField“

public static PasswordField getPasswordField() – getter metoda za „passwordField“

public static Button getLoginButton() – getter metoda za „loginButton“

2.5.5. Klasa 5 - Main

Klasa „Main“ pomoću koje se pokreće aplikacija

2.5.5.1. Atributi

2.5.5.2. Metode

public static void main(String[] args) – metoda koja pokreće aplikaciju

public void start(Stage primaryStage) throws Exception – metoda koja poziva login ekran

2.5.6. Klasa 6 - MenuScene

Klasa „MenuScene“ inicijalizira frontend dio glavnog izbornika nakon logiranja

2.5.6.1. Atributi

private static Scene scene – atribut koji predstavlja fizički sadržaji aplikacije

private static Pane layout – atribut kojim se ručno može rasporediti korisničko sučelje postavljanjem svojstava položaja i veličine za svaki element

private static VBox vbox – atributi koji postavljaju elemente u vertikalni stupac

private static Label quoteLabel, welcomeLabel - non-editable atribut za „quote“ i „welcome“

private static ImageView imageView – atribut za rad sa slikama

private static Button bookingsButton, customersButton, flightsButton, planesButton, exitButton – atributi za dodavanje gumbova „Bookings“, „Customers“, „Flights“, „Planes“ i „Exit“

2.5.6.2. Metode

public static void initialize() – metoda za inicijalizaciju frontend dijela za glavni izbornik nakon logiranja

public static Scene getScene() – getter metoda za „scene“

public static Pane getLayout() – getter metoda za „layout“

public static Button getBookingsButton() – getter metoda za „bookingsButton“

public static Button getCustomersButton() – getter metoda za „customersButton“

public static Button getFlightsButton() – getter metoda za „flightsButton“

public static Button getPlanesButton() – getter metoda za „planesButton“

public static Button getExitButton() – getter metoda za „exitButton“

2.5.7. Klasa 7 - PlaneEditScene

Klasa „PlaneEditScene“ inicijalizira frontend dio pomoćnog prozora za dodavanje i uređivanje zrakoplova.

2.5.7.1. Atributi

private static Stage dialogStage – atribut koji sadrži sve objekte aplikacije

private static Scene scene – atribut koji predstavlja fizički sadržaj aplikacije

private static GridPane gridPane – atribut koji postavlja elemente unutar fleksibilne mreže redaka i stupaca

private static VBox vboxLabel, vboxText – atributi koji postavljaju elemente u vertikalni stupac

private static Label planeNameLabel, firstClassLabel, coachLabel, economyLabel, titleLabel - non-editable atribut za „Plane name“, „First class“, „Coach class“, „Economy class“ i „Title“

private static Button okButton, cancelButton – atributi za dodavanje gumbova „Ok“ i „Cancel“

private static TextField planeNameText, firstClassText, coachText, economyText – atributi za dodavanje teksta za ime zrakoplova i količinu sjedišta za klase leta

private static BorderPane borderPane – atribut koji postavlja elemente u gornji, lijevi, desni, donji i središnji položaj

private static HBox buttons – atribut koji postavlja elemente(gumbove) u vodoravni red

2.5.7.2. Metode

public static void initialize() – metoda za inicijalizaciju frontend dijela pomoćnog prozora za dodavanje i uređivanje zrakoplova

public static Stage getDialogStage() – getter metoda za „dialogStage“

public static Scene getScene() – getter metoda za „scene“

public static Button getOkButton() – getter metoda za „okButton“

public static Button getCancelButton() – getter metoda za „cancelButton“

public static TextField getPlaneNameText() – getter metoda za „planeNameText“

public static TextField getFirstClassText() – getter metoda za „firstClassText“

public static TextField getCoachText() – getter metoda za „coachText“

public static TextField getEconomyText() – getter metoda za „economyText“

2.5.8. Klasa 8 - ViewBookingScene

Klasa „ViewBookingScene“ inicijalizira frontend dio početnog zaslona koji služi za pregled, dodavanje, uređivanje i brisanje rezervacija.

2.5.8.1. Atributi

private static Scene scene – atribut koji predstavlja fizički sadržaji aplikacije
private static Pane layout – atribut kojim se ručno može rasporediti korisničko sučelje postavljanjem svojstava položaja i veličine za svaki element
private static TableView<BookingTable> table – atribut koji omogućava prikazivanje izgleda tablice unutar aplikacije
private static TableColumn<BookingTable, String> dateColumn – atribut koji prikazuje stupac tablice u kojem se nalazi datum polaska
private static TableColumn<BookingTable, String> nameColumn – atribut koji prikazuje stupac tablice u kojem se nalazi ime kupca
private static TableColumn<BookingTable, String> routeColumn – atribut koji prikazuje stupac tablice u kojem se nalazi ruta
private static Label flightLabel, departureDateLabel, departureDateObs, arrivalDateLabel - non-editable atribut za „Flight“, „Departure date“ i „Arrival date“
private static Label arrivalDateObs, categoryLabel, categoryObs, priceLabel, priceObs - non-editable atribut za „Category“ i „Price“
private static Label customerDetailsLabel, firstNameLabel, firstNameObs, lastNameLabel - non-editable atribut za „Customer details“, „First name“, „Last name“
private static Label lastNameObs, ageLabel, ageObs, passportLabel, passportObs - non-editable atribut „Age“, „Passport“
private static Label phoneNumberLabel, phoneNumberObs - non-editable atribut „Phone number“
private static VBox flightDetails – atributi koji postavljaju elemente u vertikalni stupac
private static VBox flightDetailsObs – atributi koji postavljaju elemente u vertikalni stupac
private static VBox customerDetailsVertical – atributi koji postavljaju elemente u vertikalni stupac
private static VBox customerDetailsVerticalObs – atributi koji postavljaju elemente u vertikalni stupac
private static TextField searchField – TextField atribut za traku za pretraživanje
private static Button addBookingButton – atribut/gumb za „Add“
private static Button cancelButton – atribut/gumb za „Cancel“
private static Button backButton – atribut/gumb za „Back“
private static Button editButton – atribut/gumb za „Edit“
private static HBox buttonLayout – atribut koji postavlja elemente(gumbove) u vodoravni red

2.5.8.2. Metode

public static void initialize() – metoda za inicijalizaciju frontend dijela pomoćnog prozora za dodavanje, brisanje i uređivanje rezervacija letova

public static Scene getScene() – getter metoda za „scene“

public static Pane getLayout() – getter metoda za „layout“

public static TableView<BookingTable> getTable() – getter metoda za dohvat tablice

public static Label getDepartureDateObs() – getter metoda za „departureDateObs“

public static Label getArrivalDateObs() – getter metoda za „arrivalDateObs“

public static Label getCategoryObs() – getter metoda za „categoryObs“

public static Label getPriceObs() – getter metoda za „priceObs“

public static Label getFirstNameObs() – getter metoda za „firstNameObs“

public static Label getLastNameObs() – getter metoda za „lastNameObs“

public static Label getAgeObs() – getter metoda za „ageObs“

public static Label getPassportObs() – getter metoda za „passportObs“

public static Label getPhoneNumberObs() – getter metoda za „phoneNumberObs“

public static TextField getSearchField() – getter metoda za „searchField“

public static Button getAddBookingButton() – getter metoda za „addBookingButton“

public static Button getCancelButton() – getter metoda za „cancelButton“

public static Button getBackButton() – getter metoda za „backButton“

public static Button getEditButton() – getter metoda za „editButton“

2.5.9. Klasa 9 - ViewCustomersScene

Klasa „ViewCustomerScene“ inicijalizira frontend dio početnog zaslona koji služi za pregled, dodavanje, uređivanje i brisanje kupaca.

2.5.9.1. Atributi

private static Pane layout – atribut kojim se ručno može rasporediti korisničko sučelje postavljanjem svojstava položaja i veličine za svaki element

private static Scene scene – atribut koji predstavlja fizički sadržaji aplikacije

private static TableView<Customer> table – atribut koji omogućava prikazivanje izgleda tablice unutar aplikacije

private static TableColumn<Customer, Integer> idColumn, ageColumn – atribut koji prikazuje stupac tablice u kojem se nalazi ID kupca i dob

private static TableColumn<Customer, String> firstNameColumn, lastNameColumn, passportColumn, phoneColumn – atribut koji prikazuje stupac tablice u kojem se nalazi ime, prezime, broj telefona i putovnice

private static TextField search – TextField atribut za traku za pretraživanje

private static HBox buttonLayout – atribut koji postavlja elemente(gumbове) u vodoravni red

private static Button backButton, addButton, editButton – atributi/gumbovi za „Back“, „Add“ i „Edit“

2.5.9.2. Metode

public static void initialize() – metoda za inicijalizaciju frontend dijela pomoćnog prozora za dodavanje, brisanje i uređivanje kupaca

public static Pane getLayout() – getter metoda za „layout“

public static Scene getScene() – getter metoda za „scene“

public static TableView<Customer> getTable() – getter metoda za dohvat tablice

public static TextField getSearch() – getter metoda za „search“

public static Button getBackButton() – getter metoda za „backButton“

public static Button getAddButton() – getter metoda za „addButton“

public static Button getEditButton() – getter metoda za „editButton“

2.5.10. Klasa 10 - ViewFlightsScene

Klasa „ViewFlightScene“ inicijalizira frontend dio početnog zaslona koji služi za pregled, dodavanje, uređivanje i brisanje letova/ruta.

2.5.10.1. Atributi

private static Pane layout – atribut kojim se ručno može rasporediti korisničko sučelje postavljanjem svojstava položaja i veličine za svaki element

private static Scene scene – atribut koji predstavlja fizički sadržaji aplikacije

private static TableView<FlightTable> table – atribut koji omogućava prikazivanje izgleda tablice unutar aplikacije

private static TableColumn<FlightTable, Integer> flightIDColumn, economyColumn, coachColumn, firstClassColumn – atribut koji prikazuje stupac tablice u kojem se nalazi ID leta/rute i stupci za broj sjedišta Economy, Coach i First Class sjedišta

private static TableColumn<FlightTable, String> departureDateColumn, departureCityColumn, arrivalCityColumn – atribut koji prikazuje stupac tablice u kojem se nalazi datum polaska, grad polaska i grad dolaska

private static TableColumn<FlightTable, Double> priceColumn – atribut koji prikazuje stupac tablice u kojem se nalazi cijena karte

private static TextField search – TextField atribut za traku za pretraživanje

private static HBox buttonLayout – atribut koji postavlja elemente(gumbove) u vodoravni red

private static Button backButton, addButton, editButton, exportFlightsButton – atributi/gumbovi za „Back“, „Add“, „Edit“ i „Export“

2.5.10.2. Metode

public static void initialize() – metoda za inicijalizaciju frontend dijela pomoćnog prozora za dodavanje, brisanje i uređivanje kupaca

public static Pane getLayout() – getter metoda za „layout“

public static Scene getScene() – getter metoda za „scene“

public static TableView<FlightTable> getTable() – getter metoda za dohvat tablice

public static TextField getSearch() – getter metoda za „search“

public static Button getBackButton() – getter metoda za „backButton“

public static Button getAddButton()– getter metoda za „addButton“

public static Button getEditButton() – getter metoda za „editButton“

public static Button getExportFlightsButton()– getter metoda za „Export“

2.5.11. Klasa 11 - ViewPlaneScene

Klasa „ViewPlaneScene“ inicijalizira frontend dio početnog zaslona koji služi za pregled, dodavanje, uređivanje i brisanje zrakoplova.

2.5.11.1. Atributi

private static TableView<Plane> table – atribut koji omogućava prikazivanje izgleda tablice unutar aplikacije

private static TableColumn<Plane, Integer> planeIDColumn – atribut koji prikazuje stupac tablice u kojem se nalazi ID zrakoplova

private static TableColumn<Plane, String> planeNameColumn– atribut koji prikazuje stupac tablice u kojem se nalazi naziv zrakoplova

private static TableColumn<Plane, Integer> firstClassColumn– atribut koji prikazuje stupac tablice u kojem se nalazi broj sjedišta First Class

private static TableColumn<Plane, Integer> coachColumn– atribut koji prikazuje stupac tablice u kojem se nalazi broj sjedišta Coach Class

private static TableColumn<Plane, Integer> economyColumn– atribut koji prikazuje stupac tablice u kojem se nalazi broj sjedišta Economy Class

private static Button addButton, backButton, editButton, exportPlaneButton – atributi/gumbi za „Add“, „Back“, „Edit“, „Export“

private static VBox buttonsVbox – atributi koji postavljaju elemente u vertikalni stupac

private static Pane layout – atribut kojim se ručno može rasporediti korisničko sučelje postavljanjem svojstava položaja i veličine za svaki element

private static Scene scene – atribut koji predstavlja fizički sadržaji aplikacije

2.5.11.2. Metode

public static void initialize() – metoda za inicijalizaciju frontend dijela pomoćnog prozora za dodavanje, brisanje i uređivanje zrakoplova

public static TableView<Plane> getTable() – getter metoda za dohvat tablice

public static Button getAddButton() – getter metoda za „addButton“

public static Button getBackButton() – getter metoda za „backButton“

public static Button getEditButton() – getter metoda za „editButton“

public static Button getExportPlaneButton() – getter metoda za „exportPlanesButton“

public static Pane getLayout() – getter metoda za „layout“

public static Scene getScene() – getter metoda za „scene“